

强化训练分册

微视频精编版

明日科技 编著

HTML5+CSS3+JavaScript从入门到精通



162集微课视频，随时可学
98个应用实例+40个实战练习
250个强化训练案例

在玩中嗨翻HTML+CSS+JavaScript
趣味解读 / 易学易用 / 学练结合
知识点讲解+实例+实战+训练闯关

海量资源，可查可练
■ 实例资源库 ■ 技术资源库
■ 测试题库系统 ■ 面试资源库

清華大學出版社

软件开发微视频讲堂

HTML5+CSS3+JavaScript
从入门到精通

（微视频精编版）

明日科技　编著

清華大學出版社

北　京

内 容 简 介

本书内容浅显易懂，实例丰富，详细介绍了 HTML5+CSS3+JavaScript 开发需要掌握的各类实战知识。

全书分为两册：核心技术分册和强化训练分册。核心技术分册共 17 章，包括 HTML 基础、文本、图像和超链接、CSS3 概述、CSS3 高级应用、表格与<div>标签、列表、表单、多媒体、HTML5 新特性、JavaScript 基础、绘制图形、文件与拖放、JavaScript 对象编程、响应式网页设计、响应式组件、课程设计——游戏公园等内容。通过学习，读者可快速开发一些中小型应用程序。强化训练分册共 15 章，通过大量源于实际生活的趣味案例，强化上机实践，拓展和提升软件开发中对实际问题的分析与解决能力。

本书除纸质内容外，配书资源包中还给出了海量开发资源，主要内容如下。

- ☑ **视频讲解：总时长 17 小时，共 162 集**
- ☑ **技术资源库：800 页技术参考文档**
- ☑ **实例资源库：400 个实用范例**
- ☑ **测试题库系统：138 道能力测试题目**
- ☑ **面试资源库：369 道企业面试真题**

本书可作为软件开发入门者的自学用书或高等院校相关专业的教学参考书，也可供开发人员查阅、参考。

图书在版编目（CIP）数据

HTML5+CSS3+JavaScript 从入门到精通：微视频精编版 / 明日科技编著. —北京：清华大学出版社，2020.5
（软件开发微视频讲堂）
ISBN 978-7-302-53610-9

Ⅰ.①H… Ⅱ.①明… Ⅲ.①超文本标记语言-程序设计 ②网页制作工具 ③JAVA 语言-程序设计 Ⅳ.①TP312.8 ②TP393.092.2

中国版本图书馆 CIP 数据核字（2019）第 173918 号

责任编辑： 贾小红
封面设计： 魏润滋
版式设计： 文森时代
责任校对： 马军令
责任印制：

出版发行： 清华大学出版社
　　网　　址： http://www.tup.com.cn，http://www.wqbook.com
　　地　　址： 北京清华大学学研大厦 A 座　　**邮　　编：** 100084
　　社 总 机： 010-62770175　　**邮　　购：** 010-62786544
　　投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
　　质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 装 者：
经　　销： 全国新华书店
开　　本： 203mm×260mm　　**印　　张：** 33.5　　**字　　数：** 887 千字
版　　次： 2020 年 6 月第 1 版　　**印　　次：** 2020 年 6 月第 1 次印刷
定　　价： 99.80 元（全 2 册）

产品编号：082571-01

前　言

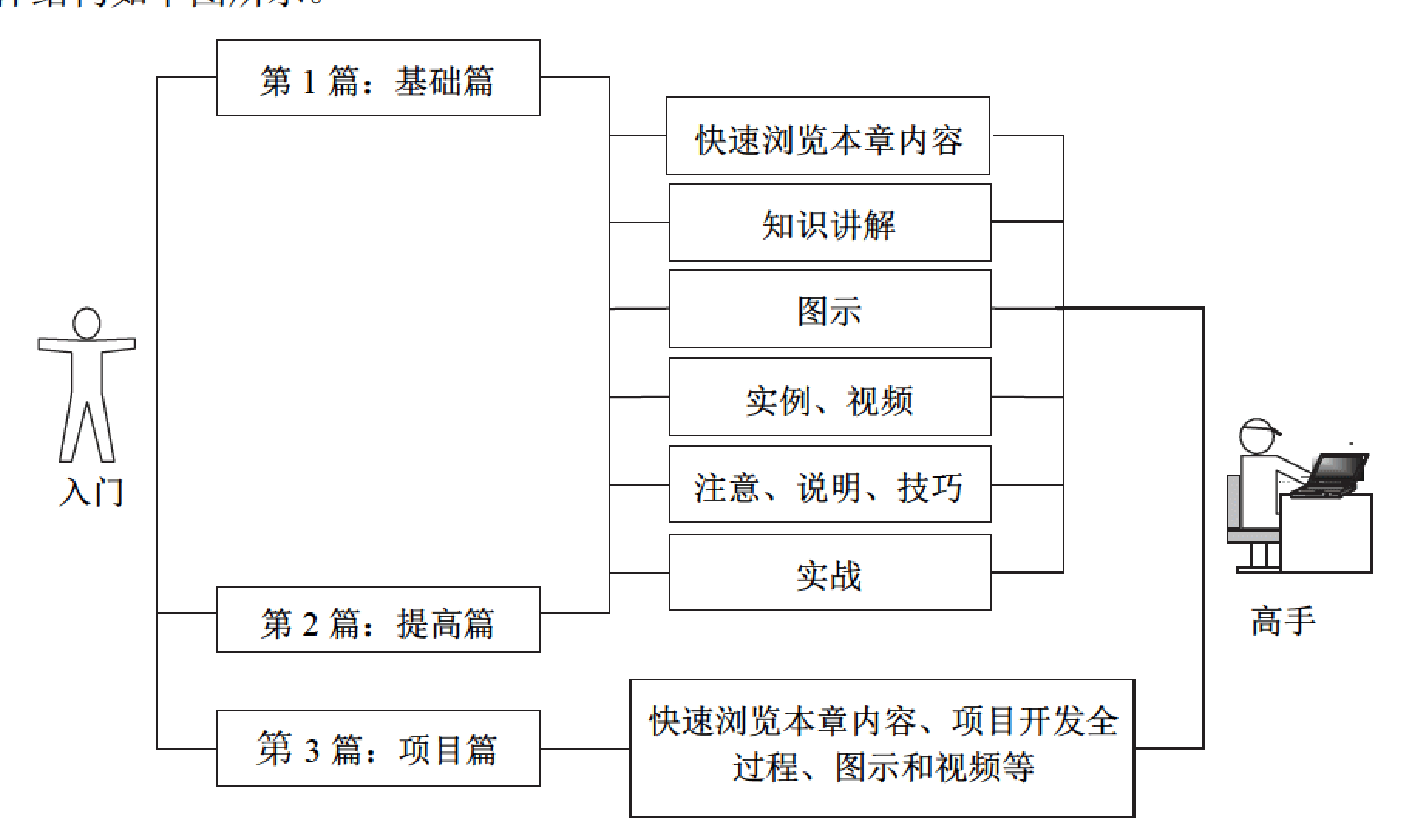
Preface

通过网站获取资讯及进行学习、娱乐、消费，已经成为人们生活和工作中不可或缺的一部分。随着技术的发展，网站内容越来越丰富，网页页面越来越美观，页面与用户的交互性也越来越强。HTML5、CSS3 和 JavaScript 的出现，使得网页设计从外观上更炫，技术上更简单。因此，HTML5+CSS3+JavaScript 这一开发组合受到广大程序员的青睐，成为网页开发人员使用的主流编程语言。

本书内容

本书分为两册：核心技术分册和强化训练分册。

核心技术分册共 17 章，提供了从入门到编程高手所必需的 HTML5、CSS3 和 JavaScript 各类核心知识，大体结构如下图所示。



第 1 篇：基础篇。本篇介绍了 HTML 基础、文本、图像和超链接、CSS3 概述、CSS3 高级应用、表格与<div>标签、列表、表单、多媒体、HTML5 新特性、JavaScript 基础等内容，使读者快速掌握开发基本内容，为以后编程奠定坚实的基础。

第 2 篇：提高篇。本篇介绍了绘制图形、文件与拖放、JavaScript 对象编程、响应式网页设计、响应式组件等内容。学习完本篇，能够开发一些中小型应用程序。

第 3 篇：项目篇。本篇通过制作一个“游戏公园”主题的网站，运用软件工程的设计思想，让读

者学习如何进行软件项目的实践开发。书中按照“需求分析→系统设计→主页实现→博客列表实现→博客详情实现→关于我们”的流程进行介绍，带领读者亲身体验开发项目的全过程。

强化训练分册共 15 章，通过 250 个来源于实际生活的趣味案例，强化上机实战，拓展和提升读者对实际问题的分析与解决能力。

本书特点

☑ **深入浅出，循序渐进**。本书以初、中级程序员为对象，先从语言基础学起（HTML 基础、CSS 基础和 JavaScript 基础），再学习网站制作常用组件的功能实现，最后学习开发一个完整项目。讲解过程步骤详尽，版式新颖，使读者在阅读时一目了然，从而快速掌握书中内容。

☑ **实例典型，轻松易学**。通过例子学习是最好的学习方式，本书通过“一个知识点、一个例子、一个结果、一段评析、一个综合应用”的模式，透彻详尽地讲述了实际开发中所需的各类知识。另外，为了便于读者阅读程序代码，快速学习编程技能，书中几乎每行代码都提供了注释。

☑ **微课视频，可听可看**。为便于读者直观感受程序开发的全过程，书中大部分章节都配备了教学微视频。这些微课可听、可看，能快速引导初学者入门，使其感受编程的快乐和成就感，进一步增强学习的信心。

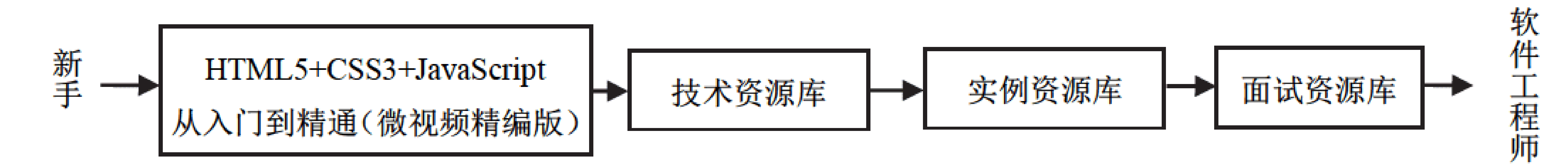
☑ **强化训练，实战提升**。软件开发学习，实战才是硬道理。核心技术分册中提供了 40 个实战练习，强化训练分册中更是给出了 250 个源自生活的真实案例。应用编程思想来解决这些生活中的难题，不但能锻炼动手能力，还可以快速提升实战技巧。如果在实现过程中遇到问题，可以从资源包中获取相应实战的源码进行解读。

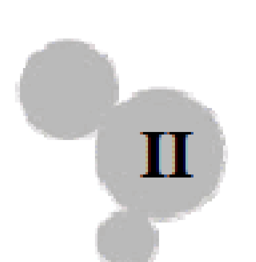
☑ **精彩栏目，贴心提醒**。本书根据需要在各章安排了很多“注意”“说明”“技巧”等小栏目，让读者可以在学习过程中更轻松地理解相关知识点及概念，更快地掌握个别技术的应用技巧。在强化训练分册中，更设置了“▷①②③④⑤⑥”栏目，读者每亲手完成一次实战练习，即可涂上一个序号。通过反复实践，可真正实现强化训练和提升。

本书资源

为帮助读者学习，本书配备了长达 17 个小时（共 162 集）的微课视频讲解。除此以外，还为读者提供了“Java Web 开发资源库”系统，可以帮助读者快速提升编程水平和解决实际问题的能力。

本书和 Java Web 开发资源库配合学习的流程如下图所示。





Java Web 开发资源库系统的主界面如下图所示。





在学习本书的过程中，可以配合技术资源库和实例资源库的相应内容，全面提升个人综合编程技能和解决实际开发问题的能力，为成为软件开发工程师打下坚实基础。

对于数学逻辑能力和英语基础较为薄弱的读者，或者想了解个人数学逻辑思维能力和编程英语基础的用户，本书提供了数学及逻辑思维能力测试和编程英语能力测试以供练习和测试。

面试资源库提供了大量国内外软件企业的常见面试真题，同时还提供了程序员职业规划、程序员面试技巧、虚拟面试系统等精彩内容，是程序员求职面试的绝佳指南。

读者对象

- ☑ 初学编程的自学者
- ☑ 编程爱好者
- ☑ 大中专院校的老师和学生
- ☑ 相关培训机构的老师和学员
- ☑ 做毕业设计的学生
- ☑ 初、中级程序开发人员
- ☑ 程序测试及维护人员
- ☑ 参加实习的“菜鸟”程序员

读者服务

学习本书时，请先扫描封底的权限二维码（需要刮开涂层）获取学习权限，然后即可免费学习书中的所有线上线下资源。本书所附赠的各类学习资源，读者可登录清华大学出版社网站（www.tup.com.cn），在对应图书页面下获取其下载方式。也可扫描图书封底的“文泉云盘”二维码，获取其下载方式。

致读者

本书由明日科技软件开发团队组织编写。明日科技是一家专业从事软件开发、教育培训以及软件开发教育资源整合的高科技公司，其编写的教材非常注重选取软件开发中的必需、常用内容，同时也很注重内容的易学、方便性以及相关知识的拓展性，深受读者喜爱。其教材多次荣获“全行业优秀畅销品种”“中国大学出版社优秀畅销书”等奖项，多个品种长期位居同类图书销售排行榜的前列。

在编写本书的过程中，我们始终本着科学、严谨的态度，力求精益求精，但错误、疏漏之处在所难免，敬请广大读者批评指正。

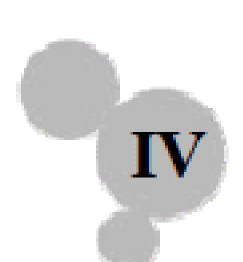
感谢您购买本书，希望本书能成为您编程路上的领航者。

“零门槛”编程，一切皆有可能。

祝读书快乐！

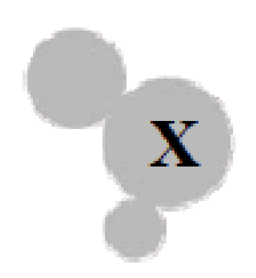
编　者

2020 年 6 月



目　录

Contents


X

第 1 章　HTML 基础



本章训练任务对应核心技术分册第 1 章“HTML 基础”部分。

重点练习内容：

1. 熟悉键盘上的字母、数字、符号的位置。
2. 使用键盘上的字符输出文字、图案等。
3. 熟悉特殊字符的输出和使用。
4. 使用字符输出常见的网站界面。
5. 使用字符输出生活中常见的一些应用场景。

应用技能拓展学习

1. WebStorm 常用快捷键

WebStorm 编辑器被广大开发者誉为 Web 前端开发神器、强大的 HTML5 编辑器、智能 JavaScript IDE 等。WebStorm 中有很多实用的快捷键，如表 1.1 所示。熟练运用 WebStorm 中的快捷键，开发过程会更高效、快捷。

表 1.1　WebStorm 常用快捷键

快　捷　键	含　　义
Ctrl+/ 或 Ctrl+Shift+/	注释（// 或者/*…*/ ）
Shift+F6	重构-重命名
Ctrl+X	删除行
Ctrl+D	复制行
Ctrl+G	查找行
Ctrl+Shift+Up/Down	代码向上/下移动
F2 或 Shift+F2	高亮显示错误或快速定位警告
写代码，按 Tab	生成代码

2. HTML 注释快捷键（WebStorm 编辑器）

（1）注释标签<!--…-->

注释标签<!--…-->用来在代码中插入注释。注释能帮助他人快速读懂代码，减轻网站维护难度。注释的内容不会在浏览器中显示。代码注释的语法格式如下：

```
<!--这是一个注释，注释在浏览器中不会显示-->
<p>这是一个段落</p>
```

注意：“<!”后和“>”前有两个连字符，有些浏览器对此有严格检查。

（2）注释快捷键

在 WebStorm 中，HTML 的注释快捷键为“Ctrl + Alt +？”。

3. 谷歌浏览器中的开发者工具

谷歌浏览器（Google Chrome）功能丰富、便利、快捷，市场占有率非常高。按 F12 键（或快捷键 Ctrl+Shift+I），可轻松访问浏览器搭载的开发者工具。

（1）开发者工具面板

谷歌浏览器的开发者工具还未被汉化，所以菜单等信息都是英文显示。以百度网页为例，进入百度首页后，按键盘上的 F12 快捷键，即可调出开发者工具面板。

如图 1.1 所示，最上方一栏包含了选择图标、移动设备图标和功能面板菜单。

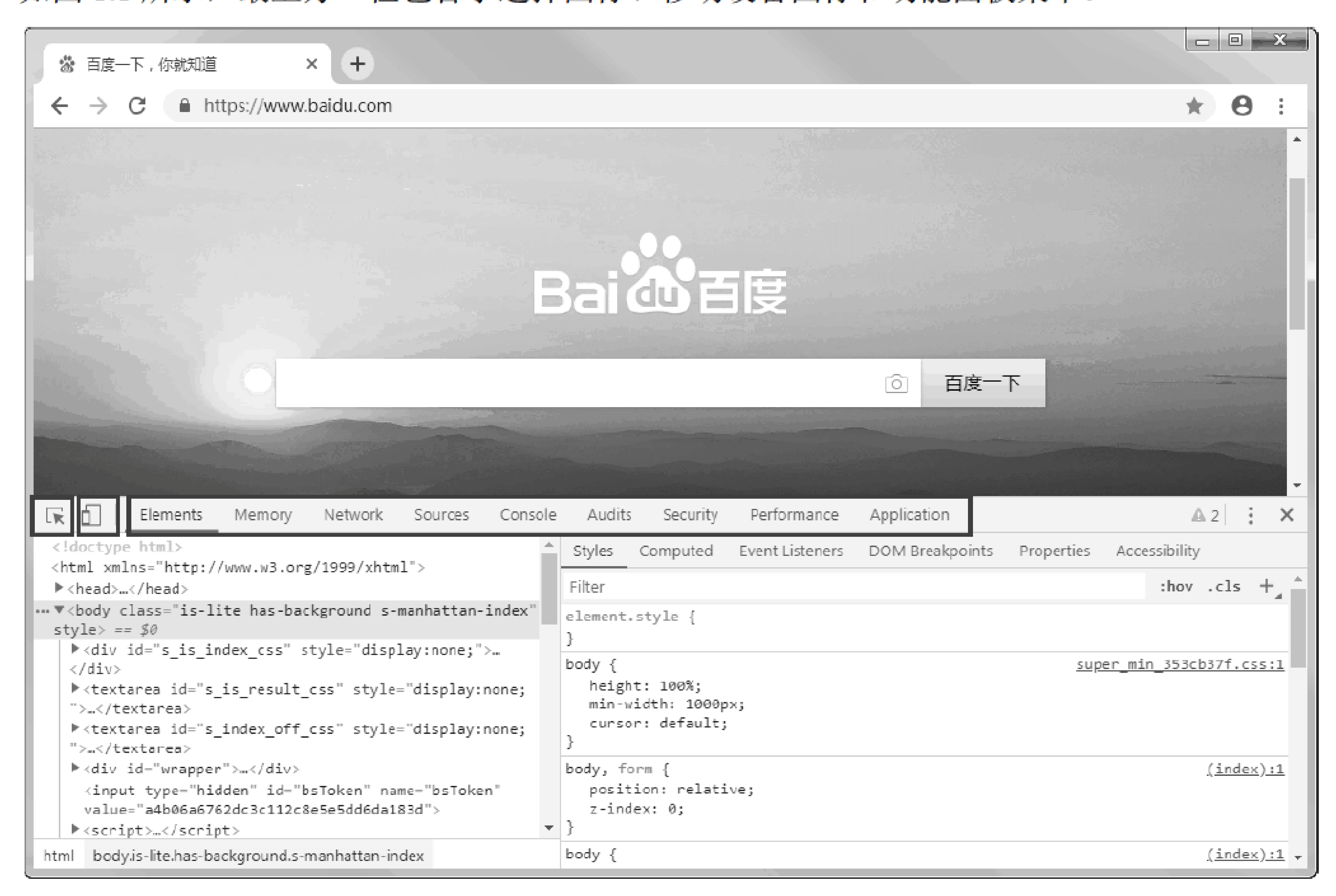


图 1.1　谷歌浏览器的开发者工具页面

☑ 选择图标 ：可使用鼠标选择网页上的元素。

☑ 移动设备图标 ：表示当前界面是否切换为移动设备模式。

☑ 功能面板菜单：包含 9 个功能面板选项，分别是 Elements、Memory、Network、Sources、Console、Audits、Security、Performance 和 Application。



（2）Elements 功能面板

Elements 面板主要用于验证标签元素及其样式，是 Web 开发者使用最多的功能选项。

如图 1.2 所示，在 Elements 功能面板中，左侧为 DOM 树信息区，可用树形图的形式显示当前的 HTML 元素信息；底部为标签层次信息栏，当前选择的 HTML 元素层次内容将显示在这里；右侧为样式信息区，当前选择的 HTML 元素的 CSS 样式等信息将显示在这里。

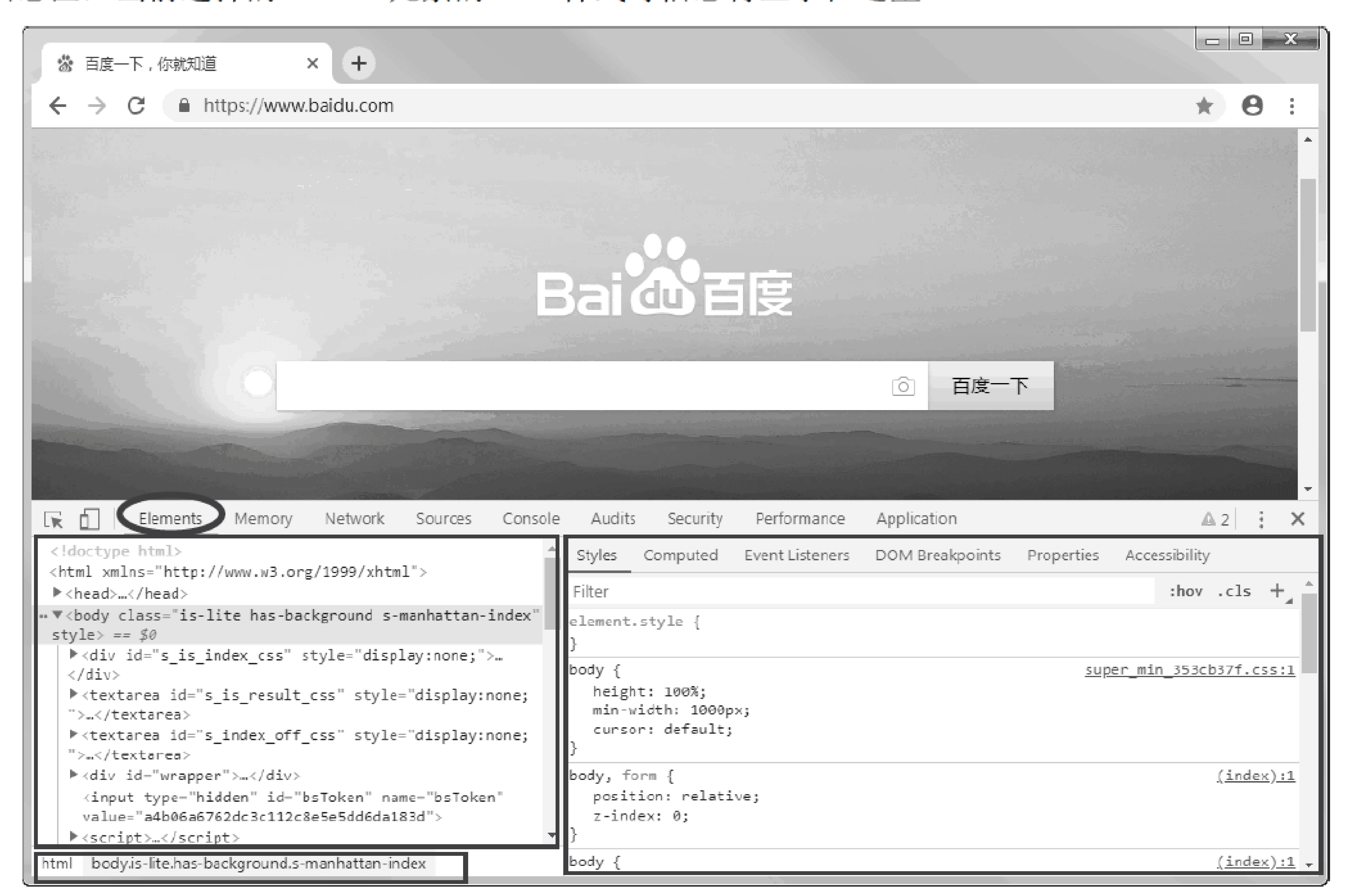


图 1.2　Element 功能区

Elements 功能区中，常用的操作有以下几种。

☑　双击标签名称，可编辑标签，如图 1.3 所示。

```
▶<div class="s_bdbri s_bdbriimg" style="opacity: 1; min-
 height: 600px; display: none;">…</div>
 </div>
▶<style>…</style>
 <div class="clear"></div>
▼<div id="head_wrapper" class="s-isindex-wrap head_wrapper s-
title-img  s-ps-islite"> == $0
   <link rel="stylesheet" href="https://ss1.bdstatic.com/
   5eN1bjq8AAUYm2zgoY3K/r/www/cache/static/protocol/https/
   soutu/css/soutu.css" type="text/css" data-for="result">
```

图 1.3　编辑标签功能

☑　双击属性，可编辑属性，如图 1.4 所示。

```
▼<div id="head_wrapper" class="s-isindex-wrap head_wrapper s-
title-img  s-ps-islite">
...    <link rel="stylesheet"
       href="https://ss1.bdstatic.com/5eN1bjq8AAUYm2zgoY3K/r/www/c
       ache/static/protocol/https/soutu/css/soutu.css" type="text/
       css" data-for="result"> == $0
     ▼<div id="s_fm" class="s_form">
       ▼<div class="s_form_wrapper soutu-env-nomac soutu-env-
       newindex" id="s_form_wrapper">
```

图 1.4　编辑属性功能

☑　右键单击，在弹出的快捷菜单中选择 Edit as HTML 命令，可再次编辑 HTML，如图 1.5 所示。



图 1.5　再次编辑 HTML

实战技能强化训练

训练一：基本功强化训练

1．输出雷军的名言　　▷①②③④⑤⑥

编写程序，输出雷军的名言“天道不一定酬勤，深度思考比勤奋工作更重要”，实现效果如图 1.6 所示。



2. 输出中英文版的“时间不等人”　▷①②③④⑤⑥

编写程序，输出中英文版的“时间不等人”及“Time and tide wait for no man”，效果如图 1.7 所示。

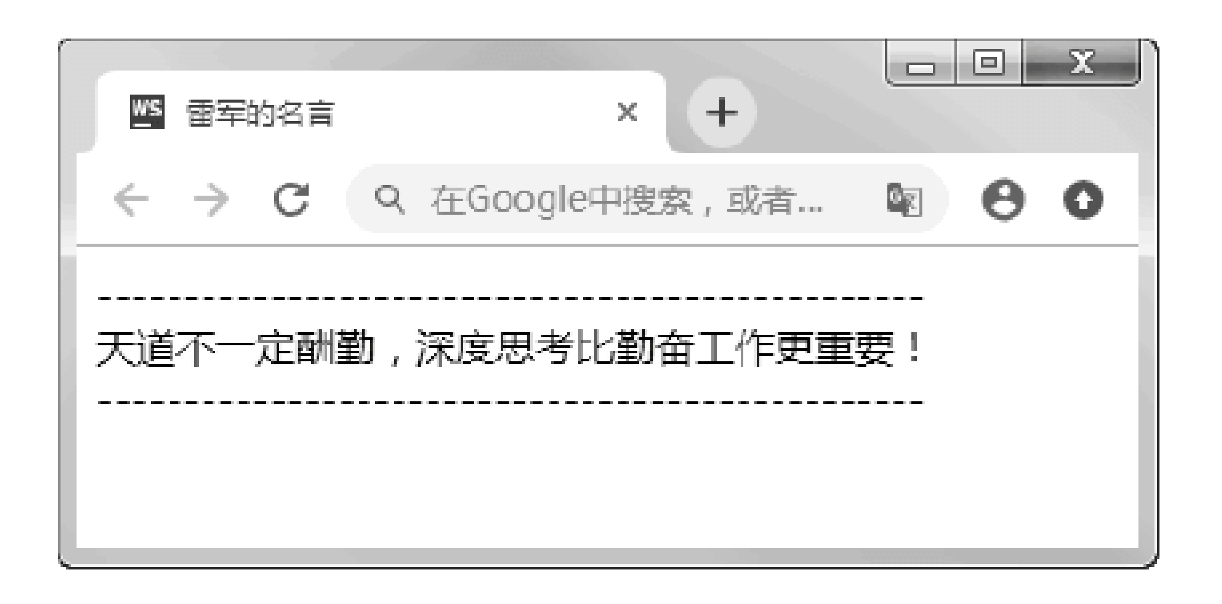


图 1.6　输出雷军的名言



图 1.7　中英文版的“时间不等人”

3. 输出俞敏洪老师的图书信息　▷①②③④⑤⑥

编写程序，输出俞敏洪老师图书《愿你的青春不负梦想》的出版信息，如图 1.8 所示。

4. 输出绕口令　▷①②③④⑤⑥

编写程序，输出绕口令《画凤凰》的内容，实现效果如图 1.9 所示。



图 1.8　输出俞敏洪老师图书的信息



图 1.9　输出绕口令

5. 输出跨年演讲主题　▷①②③④⑤⑥

2017 年跨年夜，罗振宇老师开始了他人生中第三次“时间的朋友”跨年演讲。编写程序，使用键盘上的“&”符号，输出跨年演讲主题“哪来直接登顶的人生，只有不断迭代的历程”，效果如图 1.10 所示。

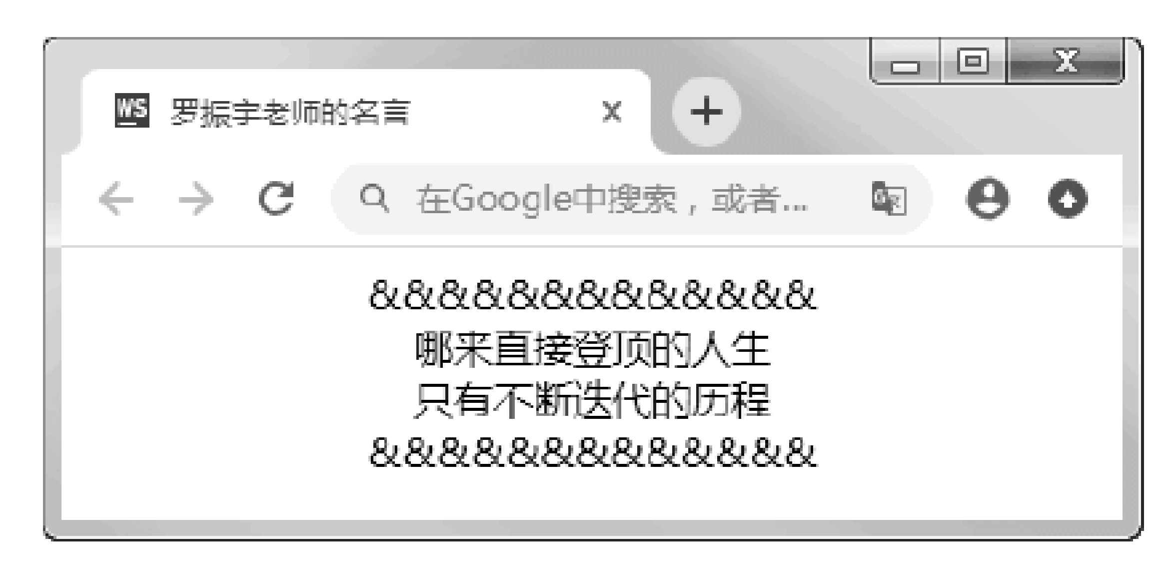


图 1.10　输出跨年演讲主题



训练二：实战能力强化训练

6. 输出淘宝网底部菜单 ▷①②③④⑤⑥

仔细观察，可发现淘宝网首页的底部有许多推广链接，还有一个导航菜单，以及各类许可证信息。编写程序，输出淘宝网底部导航菜单中的内容（如图 1.11 所示），最后实现效果如图 1.12 所示。

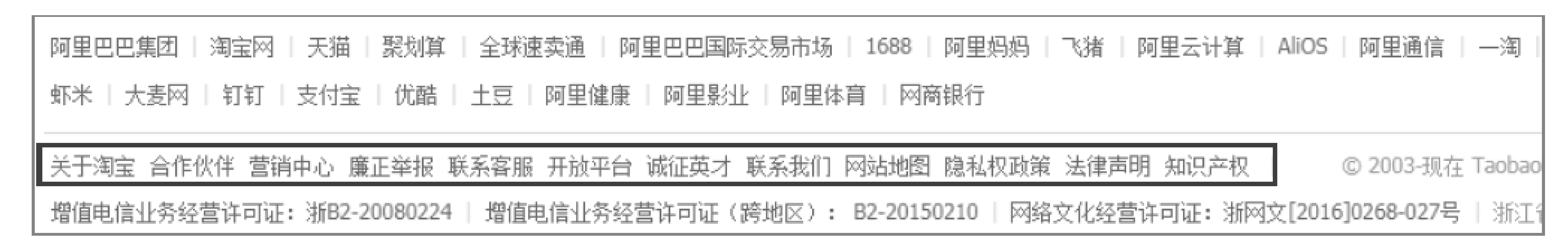


图 1.11 输出淘宝网底部菜单信息

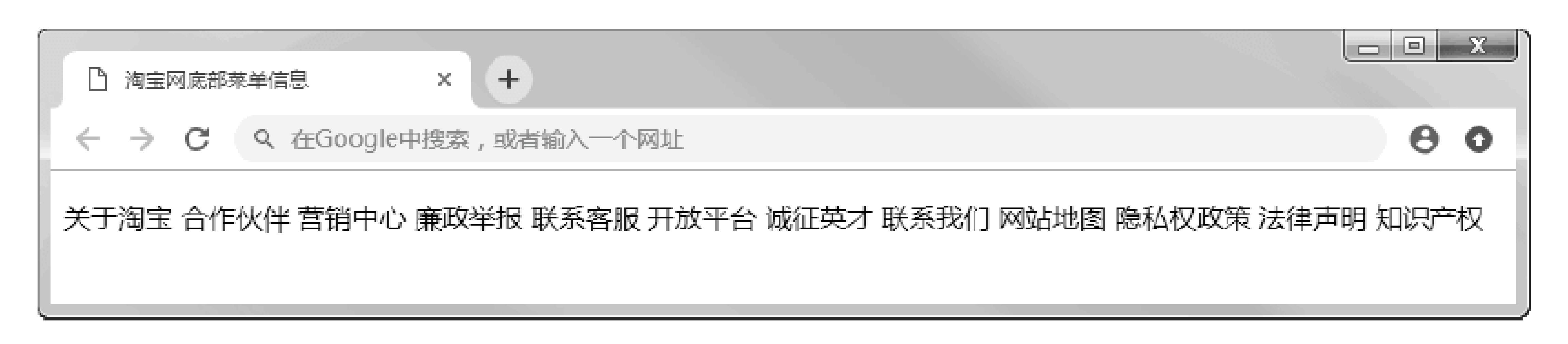


图 1.12 实现效果

7. 输出腾讯免费直播课公告 ▷①②③④⑤⑥

腾讯课堂有许多免费直播课，爱学习的人可随时选择自己感兴趣的课程进行在线学习。如图 1.13 所示为“Python 爬虫实战”的直播课程页面，右侧会以列表形式公告正在直播中的课程信息，以及即将开播的课程信息。

编写程序，模拟输出腾讯免费直播课的公告信息，效果如图 1.14 所示。



图 1.13 腾讯课堂直播课程





图1.14　输出腾讯直播课公告

8．输出明日学院简介　▷①②③④⑤⑥

在明日学院的官方网站上，可以找到相关的介绍内容，如图 1.15 所示。编写程序，输出明日学院网页中“关于我们”这段内容，如图 1.16 所示。



图1.15　明日学院官方网站



图1.16　实现效果

9．输出新浪官网底部版权信息　▷①②③④⑤⑥

网页的最下方通常是一行版权信息文字，如图 1.17 所示为新浪网页面底部的版权信息。



使用键盘上的字母或符号，编程输出如图 1.18 所示的版权信息效果。

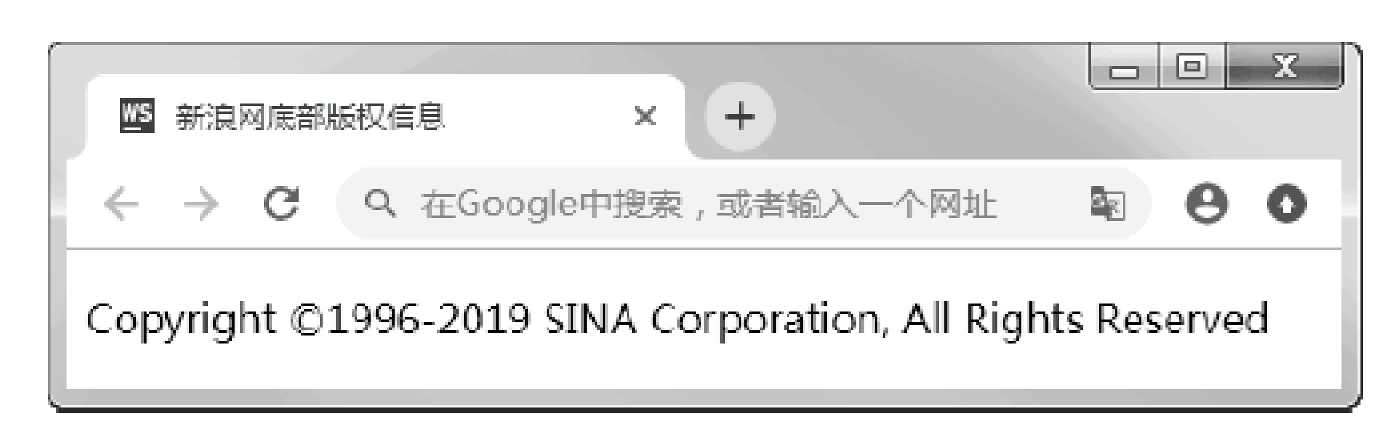


图1.17　新浪网底部版权信息

新浪网底部版权信息

在Google中搜索，或者输入一个网址

Copyright ©1996-2019 SINA Corporation, All Rights Reserved

图 1.18　实现效果

第 2 章　文　　本



本章训练任务对应核心技术分册第 2 章“文本”部分。

重点练习内容：

1. 段落标签的熟练掌握。
2. 使用特殊符号输出内容。
3. 输出常见的文字排版内容。
4. 使用字符输出生活中常见的一些应用场景。

应用技能拓展学习

1. <article>标签

<article>标签是 HTML5 中新增的标签，用来定义独立于文档且有意义的外部内容，如新闻内容、博客内容和论坛内容等。

<article>标签的语法格式如下：

```
<article>内容</article>
```

说明：<article>标签中的内容经常包含有<h1>、<header>、<footer>等标签。

在搭建博客或新闻内容的页面中，可以使用<article>标签，基本代码结构如下：

```
<article>
    <h1>文档标题</h1>
    <p>文章段落</p>
    <p>文章段落</p>
    <p>文章段落</p>
</article>
```

运行结果如图 2.1 所示。

2. <section>标签

<section> 标签用来定义文档中的节，如章节、页眉、页脚或文档中的其他部分。

需要注意的是：

☑ <section>不是一个专用来作为容器的标签。如果是为了设置样式或处理脚本，专用的是<div>标签。

- ☑ <section>里应该有标题标签（h1~6），但在文章中推荐使用 <article> 标签。
- ☑ 当元素内容被列在文档大纲中时，才适合用<section>元素。
- ☑ <section>的作用是对页面上的内容进行分块，如含标题的版块、功能区等，或对文章进行分段。不要与有自己完整、独立内容的<article>标签相混淆。

<section> 标签的语法格式如下：

```
<section>内容</section>
```

📖**说明：**< section >标签中的内容经常包含有<h1>标签。

在搭建博客或新闻内容的页面中，可以使用<article>标签，代码如下：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>section 标签</title>
</head>
<body>
<section>
    <h2>体育板块</h2>
    <p>介绍国内外最新体育信息内容</p>
</section>
<article>
    <h2>评论：中国足球何时能走出国门？</h2>
    <section>
        <h3>段落 1</h3>
        <p>段落 1 的内容</p>
    </section>
    <section>
        <h3>段落 2</h3>
        <p>段落 2 的内容</p>
    </section>
    <section>
        <h3>段落 3</h3>
        <p>段落 3 的内容</p>
    </section>
</article>
<footer>
    底部版权信息等内容
</footer>
</body>
</html>
```

运行结果如图 2.2 所示。



图 2.1　<article>标签效果



图 2.2　<section>标签效果

3. 特殊符号“ ”

特殊符号“ ”用于定义一个全角的空格。其占据的宽度正好是一个中文的宽度，而且基本上不受字体影响。

例如，在段落文本的起始处应用“ ”符号定义两个中文宽度的空格，代码如下：

```
<h2>超文本标记语言</h2>
<p>  HTML 语言（Hypertext Markup Language，中文通常称为超文本置标语言或超文本标记语言）是一种文本类、解释执行的标记语言，它是 Internet 上用于编写网页的主要语言。用 HTML 编写的超文本文件称为 HTML 文件。</p>
```

运行结果如图 2.3 所示。

超文本标记语言

　　HTML语言（Hypertext Markup Language，中文通常称为超文本置标语言或超文本标记语言）是一种文本类、解释执行的标记语言，它是Internet上用于编写网页的主要语言。用HTML编写的超文本文件称为HTML文件。

图 2.3　段首定义两个空格

实战技能强化训练

训练一：基本功强化训练

1. 输出每日励志名言 ▷①②③④⑤⑥

编写程序，应用标题标签输出每日励志名言，实现效果如图 2.4 所示。

2. 输出影片基本信息 ▷①②③④⑤⑥

《我是传奇》是威尔·史密斯主演的一部科幻电影。编写程序，使用标题标签中的 align 属性，输出该影片的海报图片、中文名、导演和主演等信息，效果如图 2.5 所示。

每日励志名言

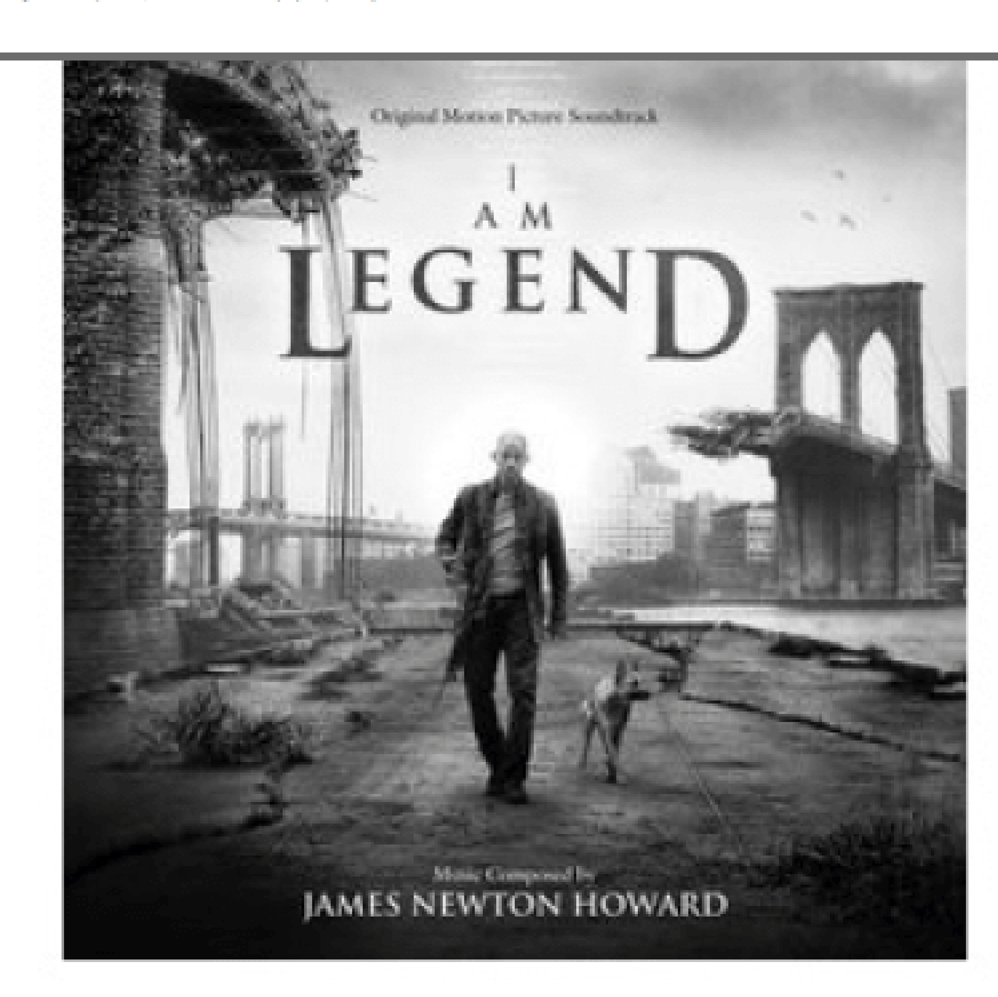
理想是人生的太阳。 —— 德莱赛

时间是一切财富中最宝贵的财富。 —— 德奥弗拉斯多

成功的秘诀，在永不改变既定的目的。 —— 卢梭

伟大的思想能变成巨大的财富。 —— 塞内加

图 2.4 输出每日励志名言



中文名：我是传奇

导演：弗朗西斯•劳伦斯

主演：威尔•史密斯

图 2.5 输出影片基本信息

3. 输出马云语录 ▷①②③④⑤⑥

编写程序，输出马云语录“别人可以拷贝我的模式，不能拷贝我的苦难，不能拷贝我不断往前的激情”。要求文字带颜色，并用如图 2.6 所示的版式输出。

4. 输出明日学院官方网站地址 ▷①②③④⑤⑥

编写程序，在明日学院简介中用斜体字输出明日学院的官方网站地址，效果如图 2.7 所示。



图 2.6　输出马云语录

明日学院简介

明日学院，是吉林省明日科技有限公司倾力打造的在线实用技能学习平台，该平台于2016年正式上线，主要为学习者提供海量、优质的课程，课程结构严谨，用户可以根据自身的学习程度，自主安排学习进度。我们的宗旨是，为编程学习者提供一站式服务，培养用户的编程思维。明日学院的官方网站地址是*http://www.mingrisoft.com*，欢迎大家访问。

图 2.7　输出明日学院简介

5. 输出 2018 年世界杯分组情况　▷①②③④⑤⑥

编写程序，输出 2018 年世界杯足球赛小组赛的分组情况，要求每组之间用一条横线进行分隔，实现效果如图 2.8 所示。

6. 输出彩色数字　▷①②③④⑤⑥

编程输出如图 2.9 所示的彩色数字，点阵用特殊符号“*”或者“□”都可以，文字颜色按实际颜色输出。

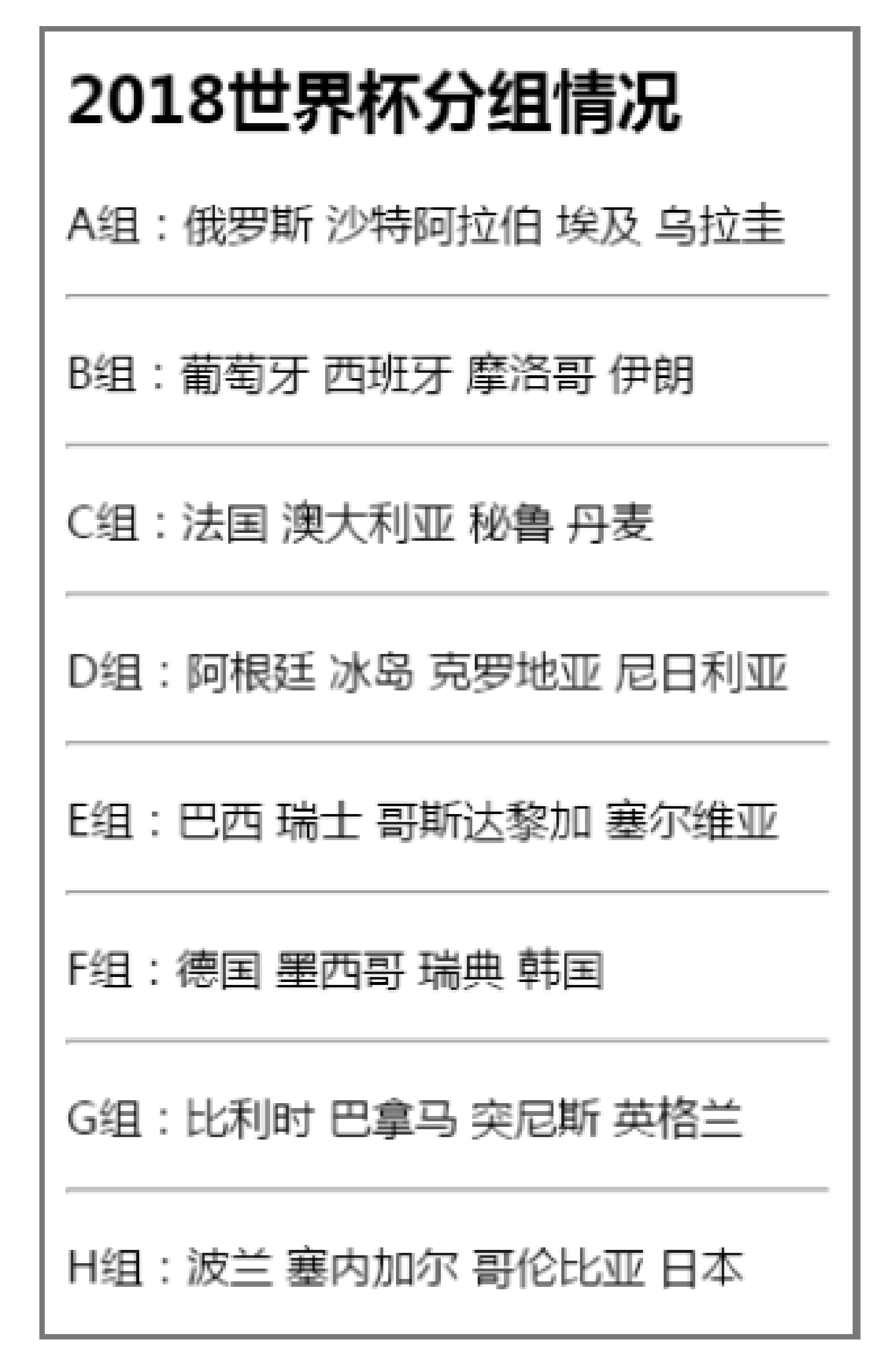


图 2.8　输出 2018 世界杯分组情况



图 2.9　输出彩色数字

7. 输出微信支付凭证 ▷①②③④⑤⑥

编写程序，输出如图 2.10 所示的微信支付凭证。不考虑文字大小，图标可以不输出，最终实现效果如图 2.11 所示。



图 2.10 微信支付凭证

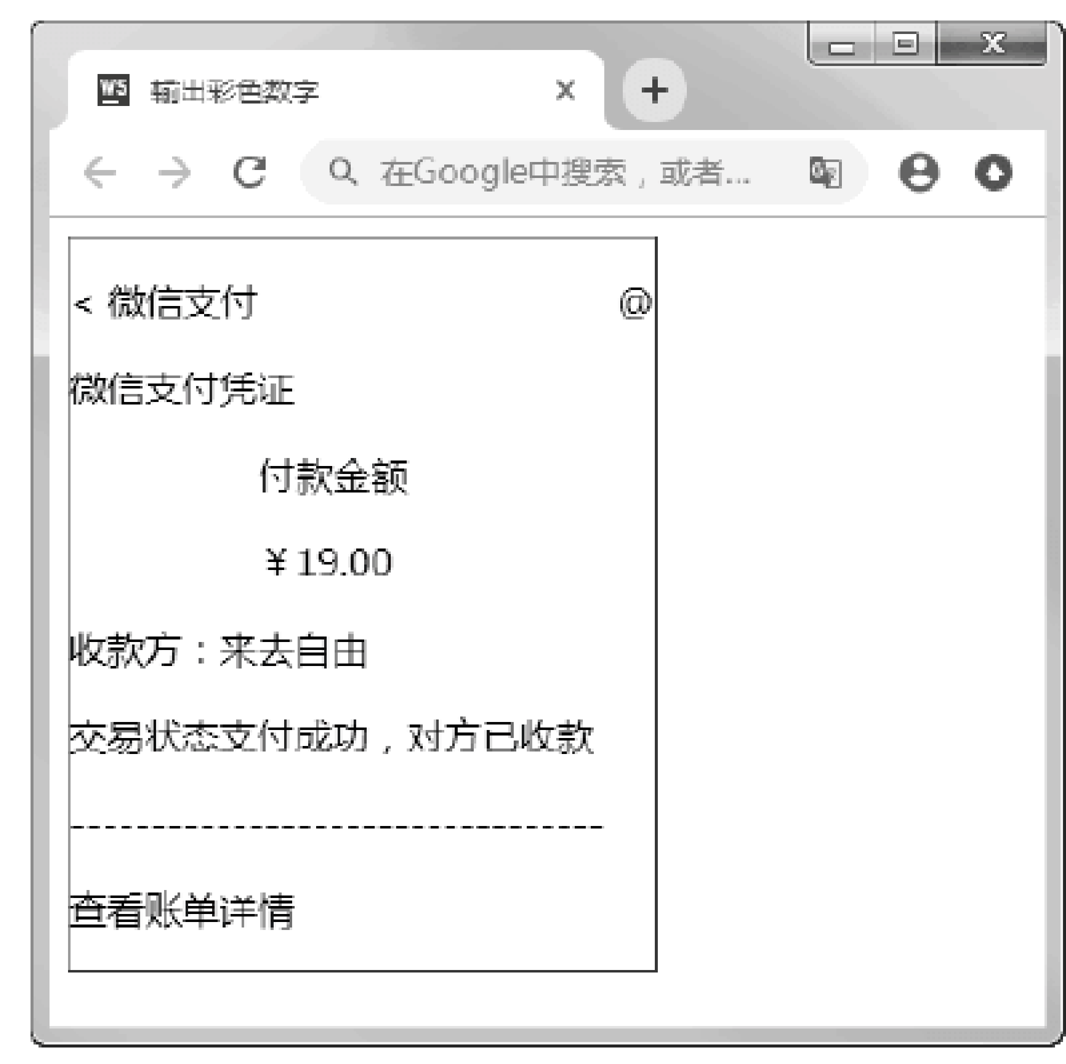


图 2.11 实现效果

8. 输出 2018 年天猫双十一总成交额 ▷①②③④⑤⑥

2018 年天猫双十一的成交额为 2135 亿元，创造了中国乃至全球商业的传奇。尝试输出如图 2.12 所示的成交额数字效果。

注意：如果计算机中安装有搜狗输入法，也可使用搜狗工具箱中的符号实现输出。



图 2.12 输出 2018 年天猫双十一总成交额

9. 输出超市购物小票 ▷①②③④⑤⑥

超市购物后，常会收到如图 2.13 所示的超市购物小票。用 HTML 编写一段程序，输出如图 2.14 所示的购物小票。（可尝试用程序输出其他各种购物清单）




操作员:1　　收据号:4370
日期:2017-09-24 17:48　　秤号:10414
商品名称
千克(个)　　元/千克(个)　　元
水晶梨（特价）
2.245　　2.56　　5.80
件数/交易数 1/1　　合计　　5.80
应付　　5.80
现金　　5.80
欢迎惠顾!

图 2.13　微信支付凭证

操作员：1　　收据号：4370
日期：2017-09-24 17:48　　称号：10414
商品名称
千克（个）　　元/千克（个）　　元
水晶梨（特价）
2.245　　2.56　　5.80
件数/交易数1/1　　合计　　5.80
应付　　5.80
现金　　5.80
欢迎惠顾!

图 2.14　实现效果

10．输出商品标价签　　▷①②③④⑤⑥

参照图 2.15，用 HTML 输出一个商品标价签。不考虑背景颜色、图标的背景图片、文字的颜色和大小，但要求整体结构完全一致。最终效果如图 2.16 所示。



图 2.15　商品标价签



图 2.16　实现效果

训练二：实战能力强化训练

11．居中排版节日板报　　▷①②③④⑤⑥

试着利用 align 属性，采用居中排版方式实现一则节日板报，效果如图 2.17 所示。

12. 居中排版唐诗　　▷①②③④⑤⑥

试着利用 align 属性的居中排版方式，输出一首唐诗，效果如图 2.18 所示。



图 2.17　节日板报

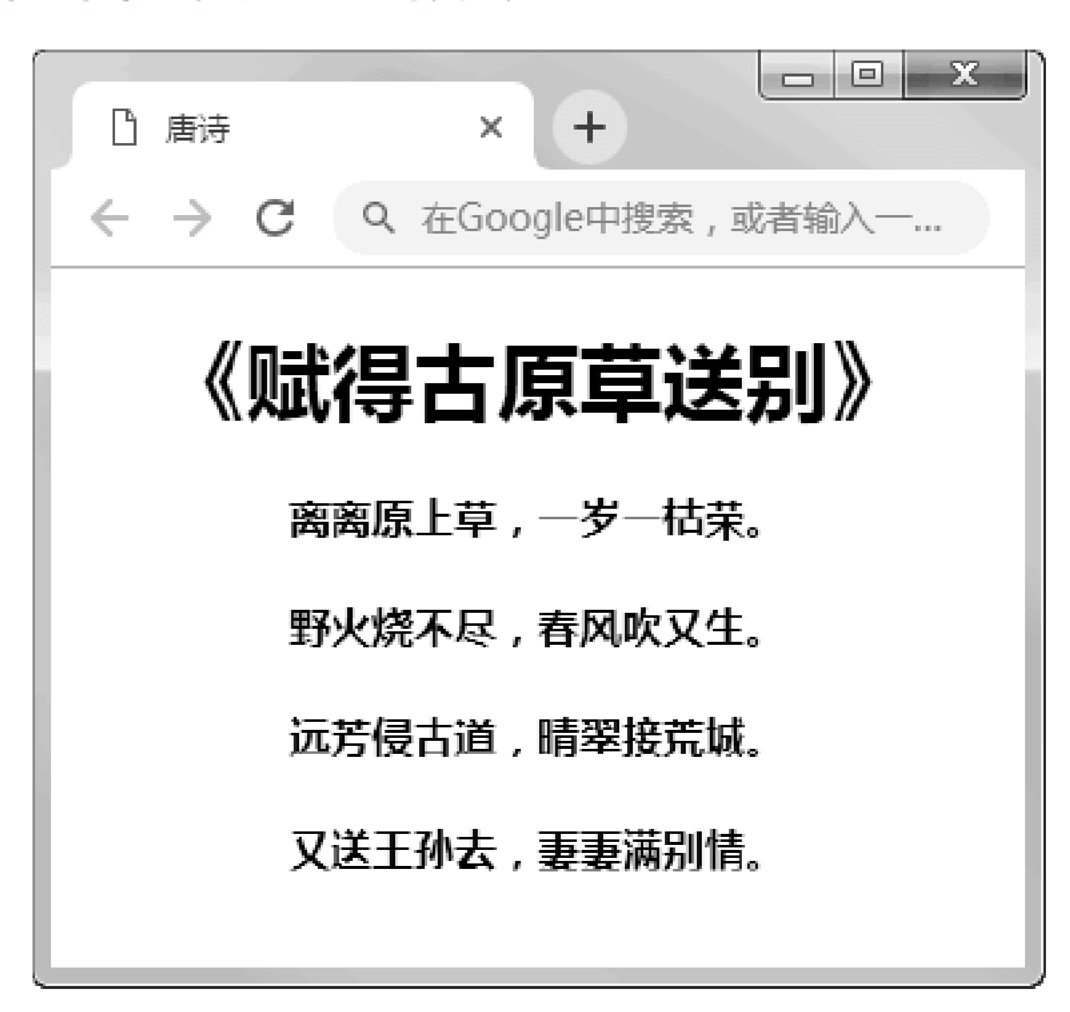


图 2.18　居中排版唐诗

13. 天气预报消息发布　　▷①②③④⑤⑥

试着使用<h1>、<h4>和<h5>标签，实现一则天气预报消息发布，效果如图 2.19 所示。

14. 打折商品清单　　▷①②③④⑤⑥

试着使用<strike>标签，完成打折商品清单页面，效果如图 2.20 所示。

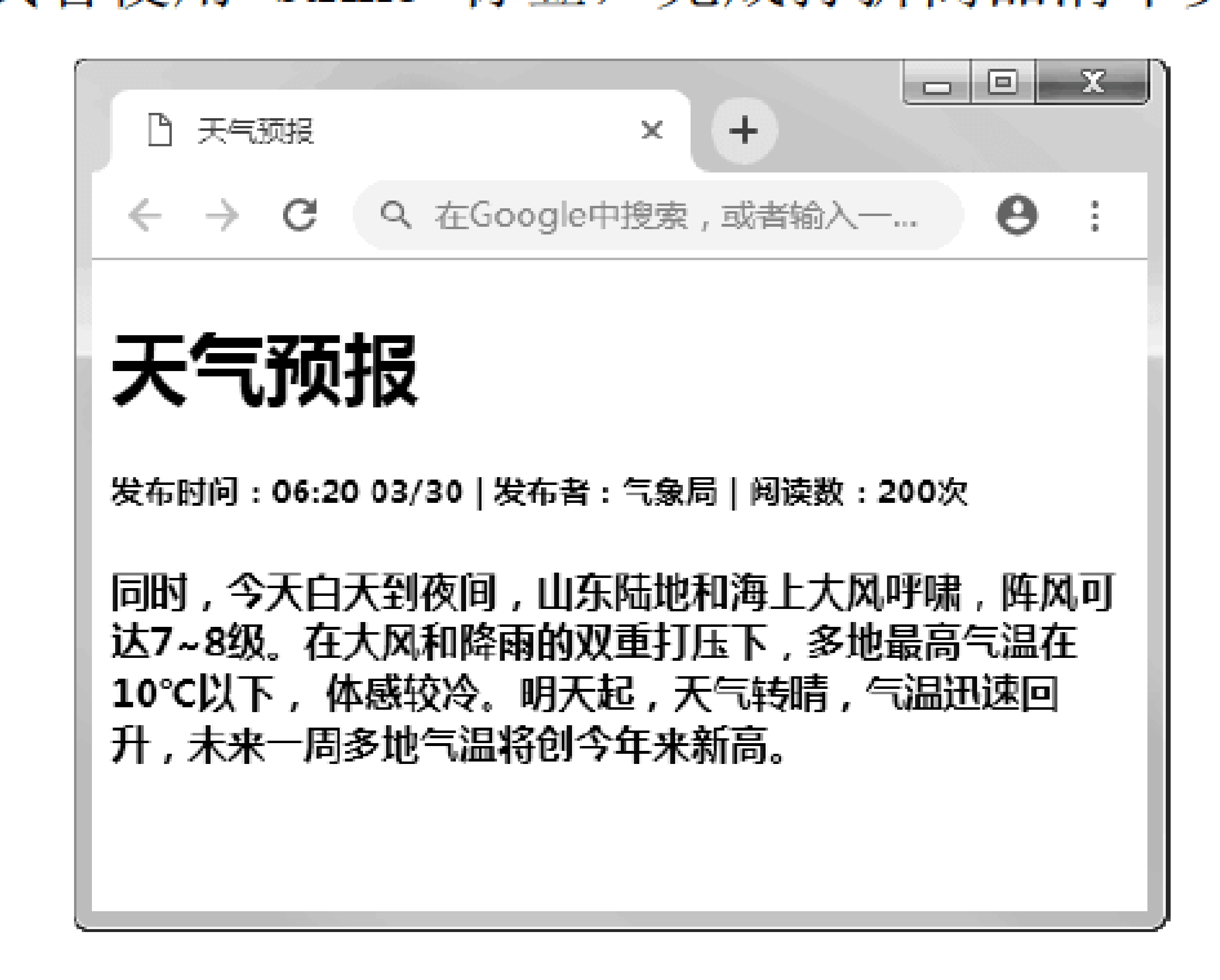


图 2.19　天气预报消息发布

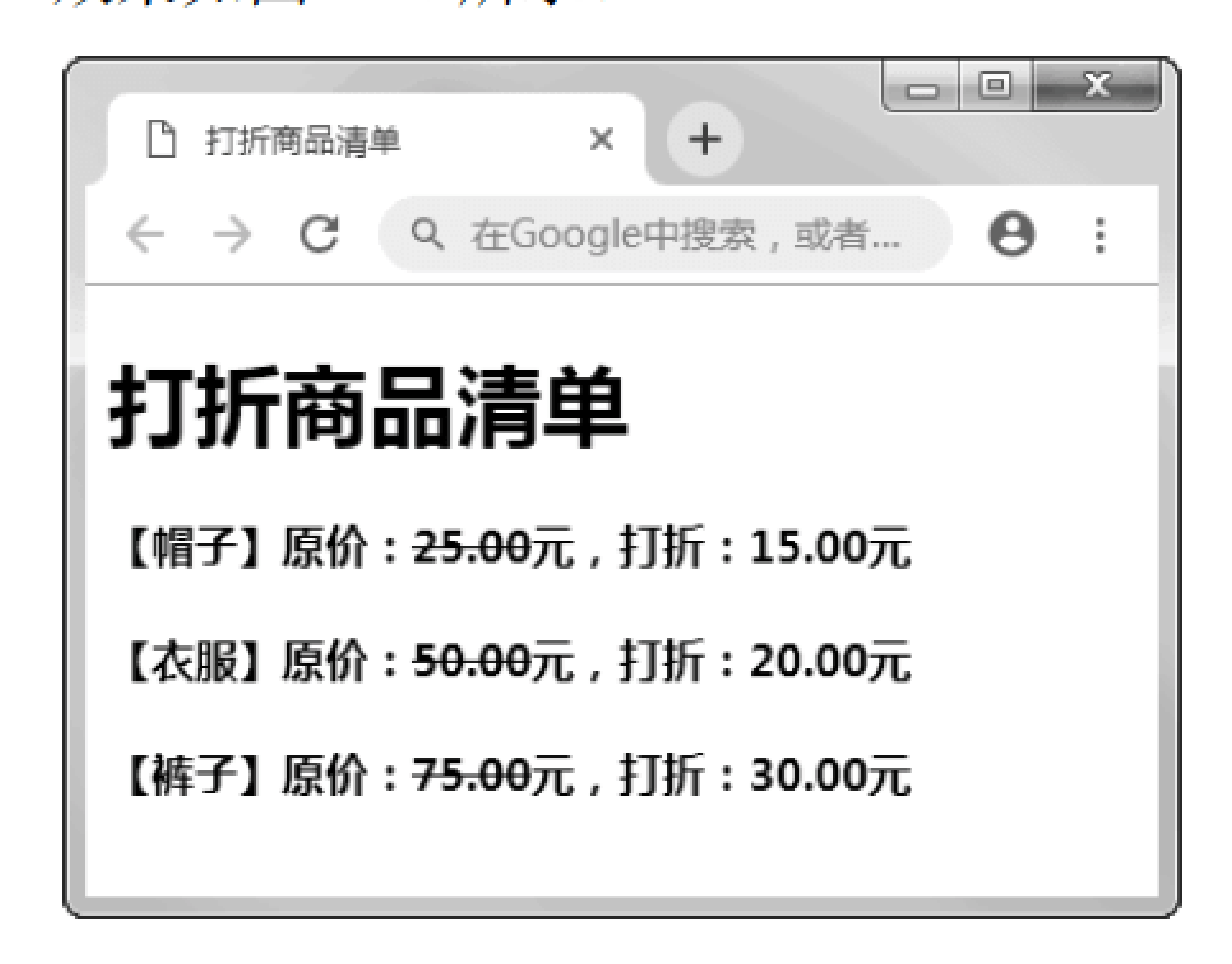
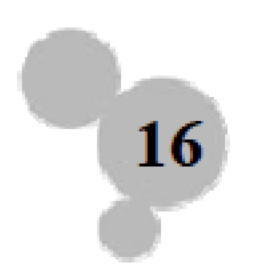


图 2.20　打折商品清单

15. 输出方程 $2X+X^2=16$　　▷①②③④⑤⑥

使用<sup>上标标签，输出方程 $2X+X^2=16$，如图 2.21 所示。



16. 输出方程 $8Y_1-X_3=20$　▷①②③④⑤⑥

使用<sub>下标标签，输出方程 $8Y_1-X_3=20$，如图 2.22 所示。



图 2.21　输出方程 $2X+X^2=16$



图 2.22　输出方程 $8Y_1-X_3=20$

17. 绘制情人节字符画　▷①②③④⑤⑥

试着使用特殊文字符号，发挥想象创意，完成一个情人节的字符画，如图 2.23 所示。

18. 给自己画个字符画像　▷①②③④⑤⑥

试着使用特殊文字符号，发挥想象创意，完成自己的字符画像。如图 2.24 所示为参考图像。



图 2.23　绘制情人节字符画



图 2.24　给自己画个字符画像

19. 完成一则通告内容　▷①②③④⑤⑥

试着使用<p>段落标签和
换行标签，完成一则通告内容，效果如图 2.25 所示。

20．使用水平线标签实现菜谱列表 ▷①②③④⑤⑥

试着使用<hr>水平线标签，实现一个菜谱列表，效果如图 2.26 所示。



图 2.25　完成一则通告内容



图 2.26　使用水平线标签实现菜谱列表

第 3 章　图像和超链接



本章训练任务对应核心技术分册第 3 章“图像和超链接”部分。

重点练习内容：

1. SVG图像的使用。
2. AlloyImage图像处理库的使用。
3. 复习图像和超链接的知识内容。
4. 使用图像和超链接，输出网络中常见的一些应用场景。

应用技能拓展学习

1. SVG 图像

SVG（Scalable Vector Graphics，可缩放矢量图形）是一种基于 XML 的图像格式。其他图像格式都是基于像素处理的，SVG 属于对图像的形状描述，其本质上是一个文本文件。SVG 图像体积较小，且不管放大多少倍，都不会失真。

设置 SVG 图像的语法格式如下：

```
<svg width="100%" height="100%">
    <circle id="mycircle" cx="50" cy="50" r="50" />
</svg>
```

说明：<svg>的 width 属性和 height 属性用于指定 SVG 图像在 HTML 元素中占据的宽度和高度。除了相对单位，也可以采用绝对单位（单位为像素）。如果不指定这两个属性，SVG 图像默认大小是 300 像素（宽）×150 像素（高）。

将 SVG 代码直接写在 HTML 网页中，它就成为网页 DOM 的一部分，可以直接用 DOM 进行操作，例如：

```
<svg id="mysvg"
     xmlns="http://www.w3.org/2000/svg"
     viewBox="0 0 800 600"
     preserveAspectRatio="xMidYMid meet">
    <circle id="mycircle" cx="400" cy="300" r="50"/>
    <svg>
```

上面的代码插入网页之后，就可以使用 CSS 定制样式。代码如下：

```
circle {
    stroke-width: 5;
    stroke: #f00;
    fill: #ff0;
}

circle:hover {
    stroke: #090;
    fill: #fff;
}
```

然后，可以用 JavaScript 代码操作 SVG，代码如下：

```
var mycircle = document.getElementById('mycircle');
mycircle.addEventListener('click', function(e) {
    console.log('circle clicked - enlarging');
    mycircle.setAttribute('r', 60);
}, false);
```

2. AlloyImage 图像处理库

AlloyImage（AI）是一个基于 HTML5 的 JS 图像处理库，集成了一些方便、快捷的图像处理 API。将其引用到网页中，结合 CSS3 等技术，可做出一些之前无法实现的优美效果，或提供一个用 JS 编写的在线图像处理工具等。

设计理念：

☑ 接口调用简单。

☑ 提供进一步的封装组合效果，方便调用。

☑ 基于 AI 编写的 AalloyPhoto，完整展示 AI 的能力和接口，以及调用方法 Demo 展示，更易于学习。

使用 AlloyImage 时，需要在 HTML 文件中引入该图像处理库，然后在 JavaScript 脚本代码中调用相关方法，并设置参数。常见调用方法如表 3.1 所示。

表 3.1　AlloyImage 中方法及其说明

属　性　值	含　　义
act()	对单个图层进行效果处理
add()	合并一个图层到本图层上去
show()	将当前图层的 canvas 元素显示出来
replace()	将本图像替换为 image 显示
clone()	复制图层对象
view()	预览模式下处理
excute()	执行预览模式下的操作
cancel()	取消预览模式下所做的处理
ctx	调用原生 Canvas API

例如，如果对图像进行灰度处理，需调用 act()方法，语法如下：

```
var grayPic = picTranseObj.act("灰度处理");
```

📖**说明**：在 act()方法中添加不同的参数，可实现不同的处理效果。

例如，新建 HTML 文件，添加图片并设置 ID，引入 AlloyImage API 文件，关键代码如下：

```
<div class="cont">
    <p>将鼠标悬停在图片上，可观看图片灰度处理后的效果</p>
    <img id="pic" src="img/view1.jpg" alt="" width="1000"/>
</div>
```

例如，在<head>标签中添加<style>标签，并在其中设置页面样式，关键代码如下：

```
<style type="text/css">
    .cont {
        text-align: center;
        font: bold 25px/20px "";
        margin: 40px auto;
    }
</style>
```

例如，在<div>标签下添加<script>标签，并在其中调用相关方法对图像进行灰度处理，代码如下：

```
<script type="text/javascript">
    var pics = document.getElementById("pic");
    pics.loadOnce(function () {
        var picTranseObj = psLib(this);            //创建一个 psLib 对象
        var origin = picTranseObj.clone();         //复制原始对象，作为原始副本
        var grayPic = picTranseObj.act("灰度处理");
        this.onmouseout = function () {
            origin.replace(this);
        };
        this.onmouseover = function () {
            grayPic.replace(this);
        }
    })
```

实战技能强化训练

训练一：基本功强化训练

1．制作购物网站的“促销活动”页面　▷①②③④⑤⑥

编写程序，输出如图 3.1 所示的“促销活动”页面。

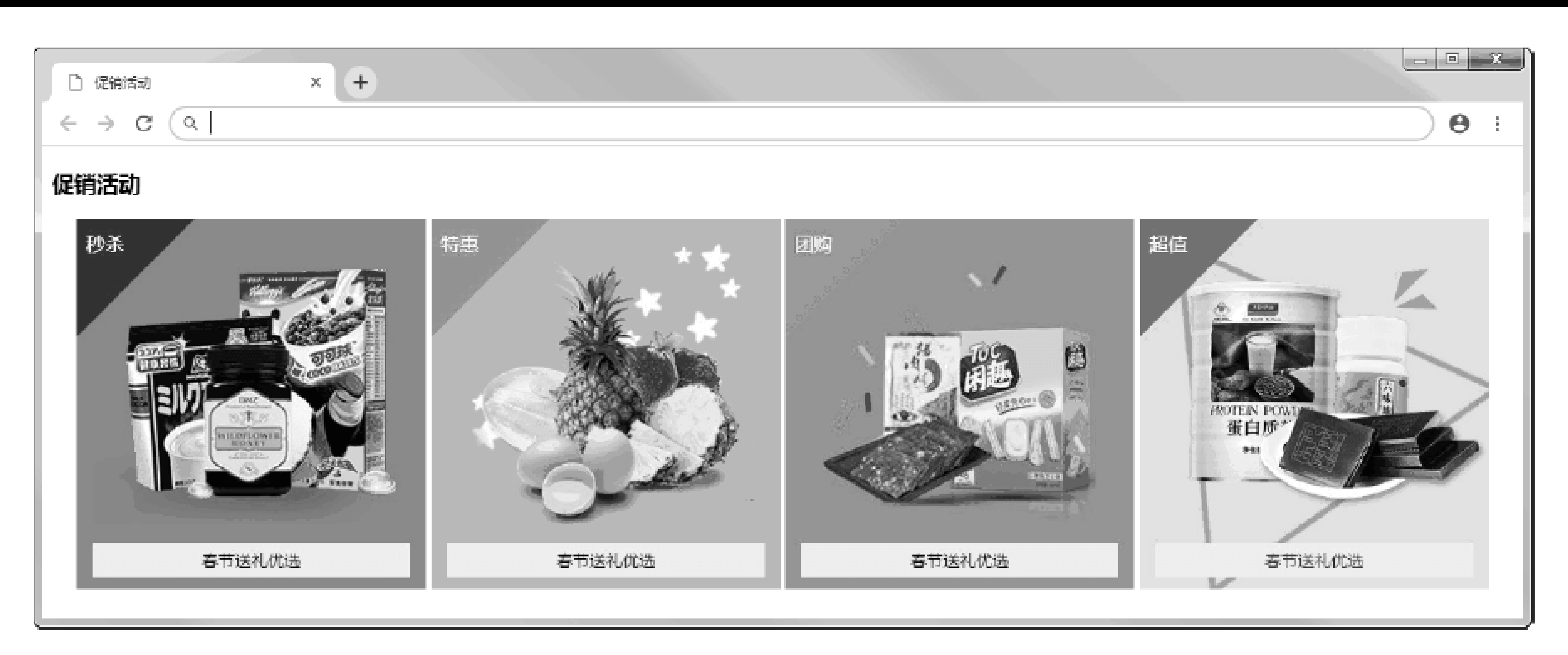


图 3.1　制作购物网站的“促销活动”页面

2. 使用图像标签在页面中显示图书封面　　▷①②③④⑤⑥

编写程序，利用图像标签实现图书展示功能，即在页面中显示多本图书的封面，如图 3.2 所示。



图 3.2　使用图像标签显示图书封面

3. 制作手机商城的商品展示页面　　▷①②③④⑤⑥

编写程序，制作手机商城的商品展示页面，要求页面的图片大小统一，如图 3.3 所示。





图 3.3　手机商城的商品展示页面

4．制作图书导航页面　▷①②③④⑤⑥

编写程序，制作图书网站首页的导航菜单和宣传图片，效果如图 3.4 所示。



图 3.4　图书网站的导航页面

5．展示玫瑰的生长过程

▷①②③④⑤⑥

编写程序，在网页中通过插入图片的方法，展示玫瑰的生长过程，效果如图 3.5 所示。



图 3.5　玫瑰的生长过程

6. 布局电脑配件页面　▷①②③④⑤⑥

编写程序，实现如图 3.6 所示的电脑配件页面效果。



图 3.6　手机商品布局页面

7. 为图书添加替换文本和提示文字　▷①②③④⑤⑥

编写程序，制作一个编程图书的介绍页面，并为图片添加替换文本和提示文字，如图 3.7 所示。



图 3.7　制作图书介绍页面

8. 制作商品评价页面，提示文字为商品信息 ▷①②③④⑤⑥

编写程序，制作商品评价页面，设置图片的提示文字为商品的相关信息，如图 3.8 所示。



图 3.8 商品评价页面

9. 使用链接制作网站的导航菜单 ▷①②③④⑤⑥

编写程序，使用文本链接，制作网站的导航菜单，效果如图 3.9 所示。



图 3.9 使用链接制作网站的导航菜单



10．“展开全文”页面链接效果　　▷①②③④⑤⑥

编写程序，使用文本链接，制作网页中“展开全文”页面，效果如图 3.10 所示。



图 3.10　“展开全文”页面链接效果

训练二：实战能力强化训练

11．仿淘宝放大镜效果　　▷①②③④⑤⑥

在电商网站的商品详情页中，常常附带放大镜功能，可以非常方便地查看商品细节。

编写程序，模拟淘宝的放大镜功能。如图 3.11 所示，当鼠标光标在左侧图片中移动时，图片中会显示一块半透明区域，半透明区域中的内容会放大并显示在右侧图片中。



图 3.11　仿淘宝放大镜效果

12. SVG 实现图片模糊效果 ▷①②③④⑤⑥

HTML5 的 SVG 包含多种滤镜，使用这些滤镜可对图片或文字设置一些特殊效果。

编写程序，使用高斯滤镜使图像变模糊。首先显示一张模糊的图片，下方提示高斯模糊值为 3，如图 3.12 所示。向右拖动滑块，将增大模糊值，图像会更加模糊。向左拖动滑块，图片将变清晰。将滑块拖至最左侧，即高斯模糊值为 0，图像将变清晰，如图 3.13 所示。



图 3.12　高斯模糊值为 3



图 3.13　高斯模糊值为 0

13. SVG 实现点赞特效 ▷①②③④⑤⑥

编写程序，通过 SVG 实现朋友圈点赞特效。打开页面，可看到一条朋友圈动态，如图 3.14 所示。单击右下方的爱心图标时，爱心会变为蓝色，同时爱心周围会出现一个圆圈，先逐渐放大，又逐渐模糊直到消失，效果如图 3.15 所示。



图 3.14　朋友圈动态



图 3.15　点赞特效

14. 应用 AlloyImage 对图像进行灰度处理 ▷①②③④⑤⑥

编写程序，对图像进行灰度处理。打开图片，原图效果如图 3.16 所示。当鼠标放置在图片上时，图片被灰度处理，变成灰白色图片，如图 3.17 所示。当鼠标从图片上离开时，图片又恢复为彩色。

注意：本任务需要在火狐浏览器或者 IE11 以上的版本中运行。使用谷歌浏览器时，需要通过 WebStorm 打开谷歌浏览器。

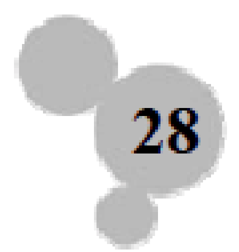




图 3.16　原图



图 3.17　图像灰度处理后的效果

15．应用 AlloyImage 对图像进行反色处理　▷①②③④⑤⑥

编写程序，对图像进行反色处理。打开图片，原图效果如图 3.18 所示。单击图片，可看到图像被反色处理后的效果，如图 3.19 所示。

注意： 本任务需要在火狐浏览器或者 IE11 以上的版本中运行。使用谷歌浏览器时，需要通过 WebStorm 打开浏览器。



图 3.18　原图



图 3.19　对图像反色处理后的效果



第 4 章　CSS3 概述

本章训练任务对应核心技术分册第 4 章“CSS3 概述”部分。

重点练习内容：

1. 理解CSS在页面中的作用。
2. 掌握CSS的基本语法、常用属性。
3. 使用CSS美化文本、图像等内容。
4. 通过CSS在页面中设置动画效果。

应用技能拓展学习

1. Velocity.js 框架

Velocity.js 是一个简单、易用、高性能、功能丰富的轻量级 JS 动画库，其官方网站下载地址是 http://velocityjs.org/。它能与 jQuery 完美协作，并和$.animate()有着相同的 API。同时，它不依赖于 jQuery，可单独使用。

Velocity 不仅包含了 $.animate() 的全部功能，还拥有颜色动画、转换动画（transforms）、循环、缓动、SVG 动画、滚动动画等特色功能。

下面的代码中就调用了 Velocity.js 框架。

```
<body>
<!-- dom -->
<div class="element"></div>
<script src="https://cdn.jsdelivr.net/npm/jquery@3.1.0/dist/jquery.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/velocity-animate@1.5.0/velocity.min.js"> </script>
<script>
    $(".element").velocity({
        left: "200px"
    }, {
```

```
            duration: 450,
            delay: 1000
        });
        // $.animate() 的写法，效果同上
        $(".element").delay(1000).velocity({left: "200px"}, 450);
    </script>
</body>
```

2. 活用 position 属性

在训练 16 中，用到了 CSS 中的 position 属性。

该实例中，通过数组存放图片轮播时 5 张图片的宽度、高度、透明度、位置以及优先级信息，当鼠标单击向左或向右按钮时，依次改变图片的大小位置以及透明度等样式。实现相册图片的切换效果。

改变透明度通过 opacity 实现，其语法如下：

```
opacity：number
```

需要说明的是，number 为指定对象的不透明度，其取值范围为 0.0～1.0。

实战技能强化训练

训练一：基本功强化训练

1. 制作一个简单的注册/登录页面　▷①②③④⑤⑥

浏览网站时，通常需要注册或登录账号。编写一个简单的注册/登录页面，并通过 CSS 美化页面内容，效果如图 4.1 所示。

2. 制作网页版个人简历　▷①②③④⑤⑥

尝试为自己制作一份网页版个人简历，然后通过 CSS 美化简历，效果如图 4.2 所示。



图 4.1　简单的注册/登录页面



图 4.2　网页版个人简历

3. 制作网页版生日贺卡　　▷①②③④⑤⑥

小张生日快到了，帮他制作一个网页版生日贺卡，效果如图 4.3 所示。

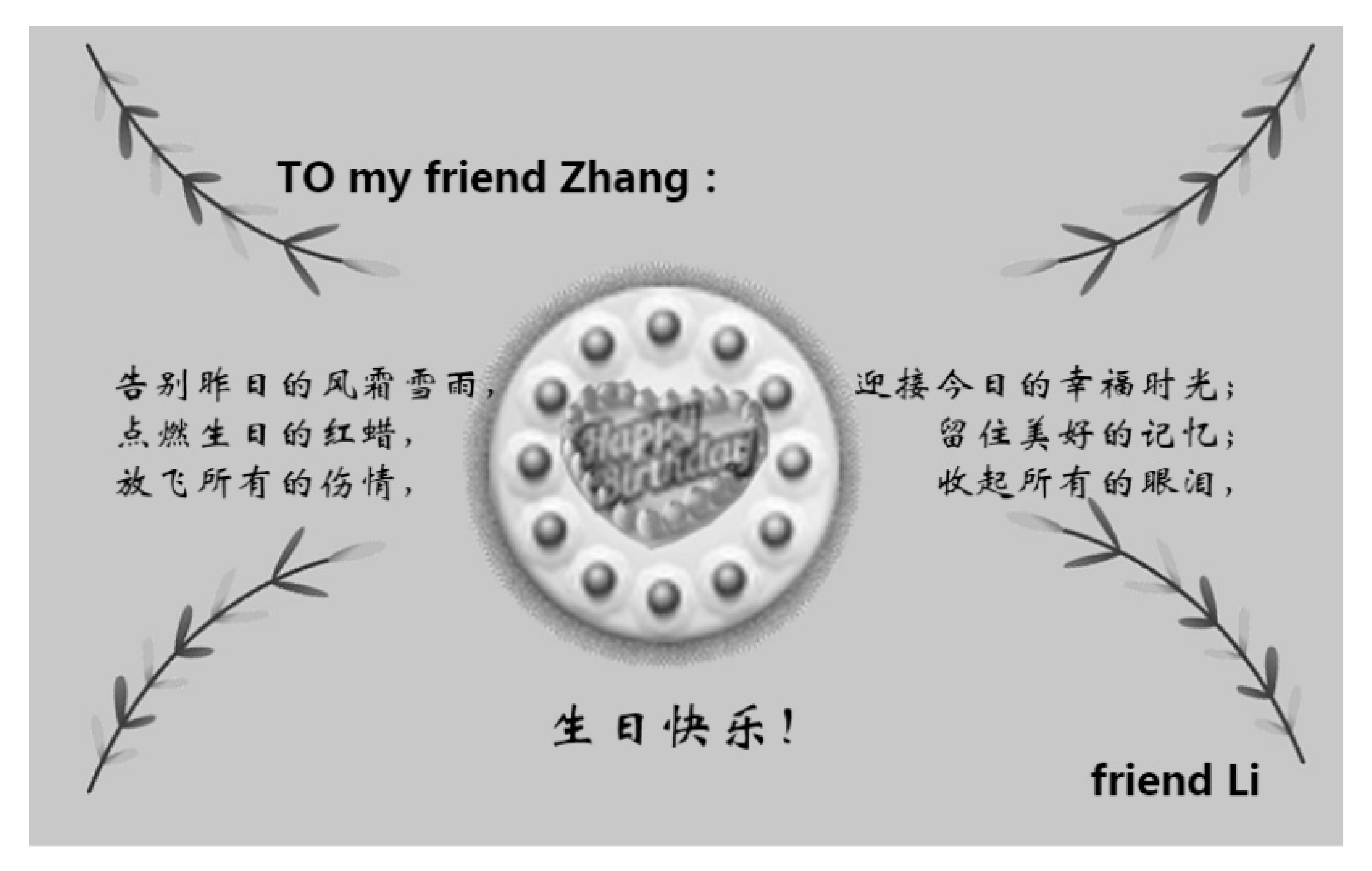


图 4.3　网页版生日贺卡

4. 制作网页版宣传海报　　▷①②③④⑤⑥

李先生新开一家手机店，为他设计一个新店开张宣传海报页面，效果如图 4.4 所示。



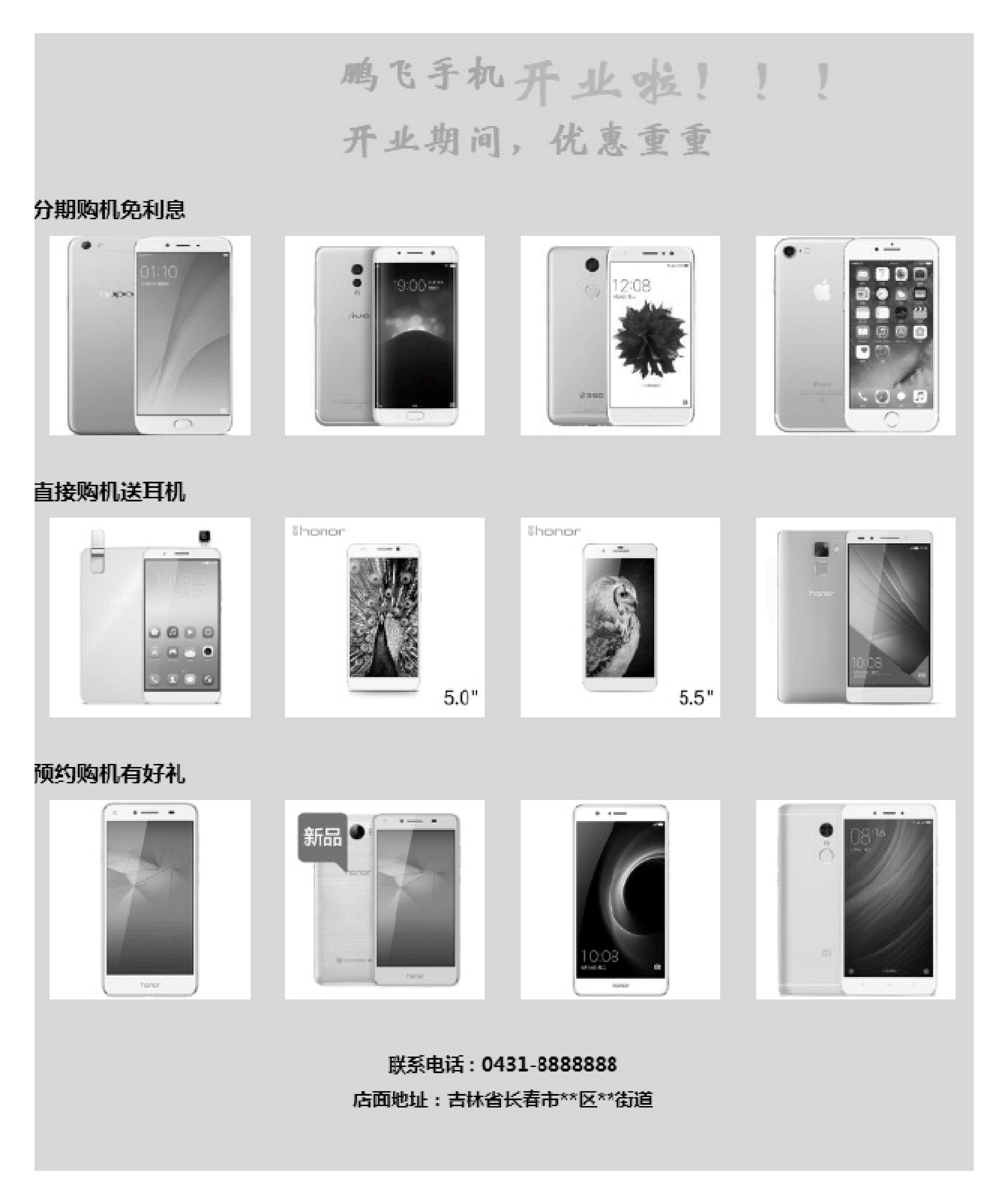


图 4.4　网页版宣传海报

5．制作购物商城的商品展示页面　▷①②③④⑤⑥

制作购物商城的商品展示页面，效果如图 4.5 所示。当鼠标放置在商品上时，图片放大。另外，需要设置不同状态下的超链接字体颜色。



图 4.5　手机展示页面

6. 制作手机介绍页面　　▷①②③④⑤⑥

制作一个手机的宣传介绍页面，效果如图 4.6 所示。

（提示：设置网页中第一行文字以及第二行第一个文字的样式）



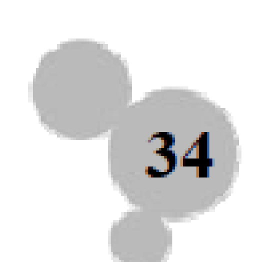
图 4.6　手机介绍页面

7. 制作个人空间主页　　▷①②③④⑤⑥

尝试编写一个个人空间主页，并使用 CSS 美化它，效果如图 4.7 所示。



图 4.7　个人空间主页



8．实现页面横幅广告效果　▷①②③④⑤⑥

编写程序，实现手机商城首页的横幅广告效果，如图 4.8 所示。
（提示：使用背景以及文本的相关属性）

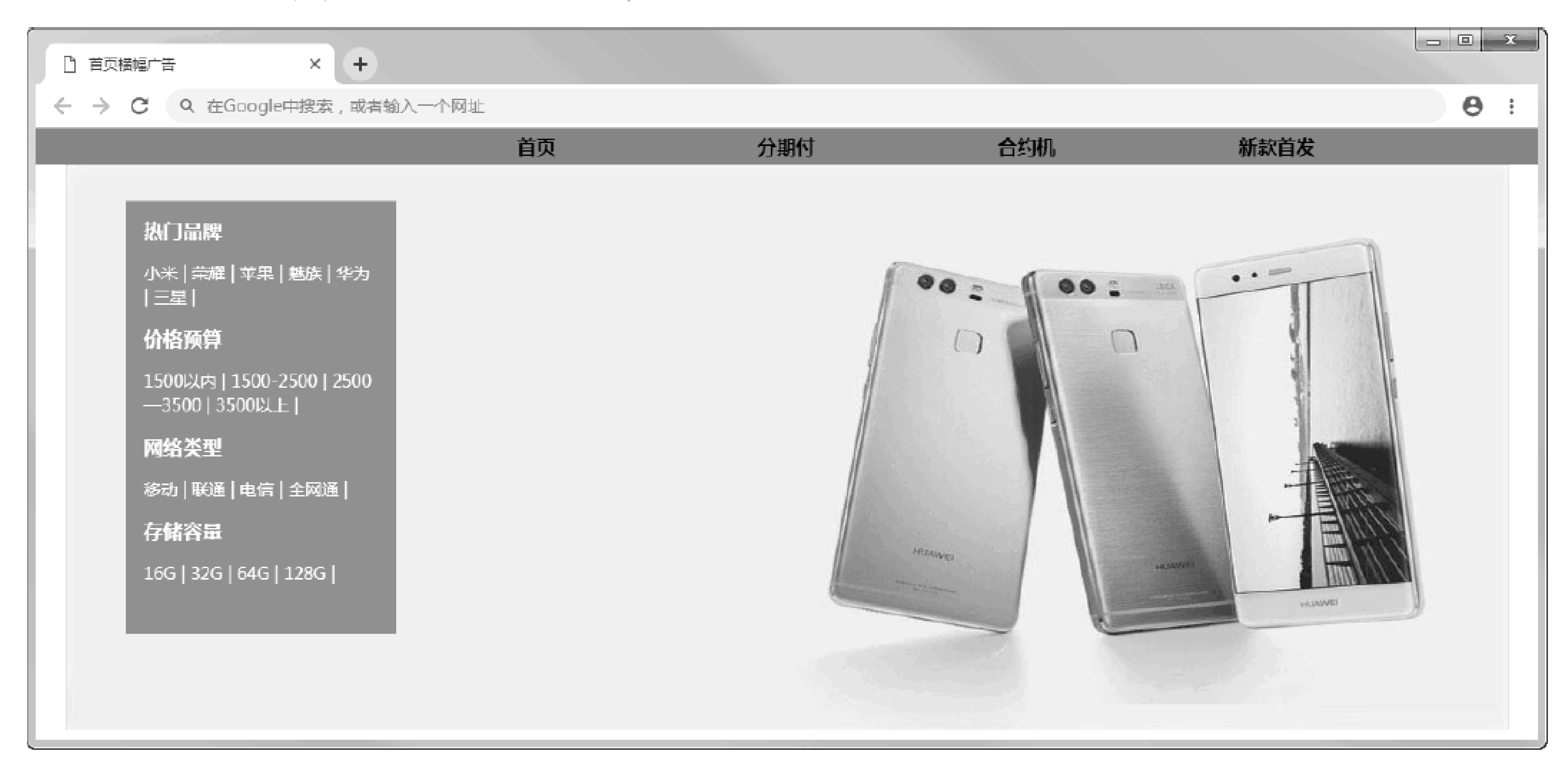


图 4.8　首页横幅广告效果

9．制作购物商城的“热销爆款”页面　▷①②③④⑤⑥

制作购物商城首页中的“热销爆款”页面，如图 4.9 所示。

换季换新机







图 4.9　“热销爆款”页面

10．制作购物商城的“精品手机”页面　　▷①②③④⑤⑥

编写程序，制作手机商城的“精品手机”页面，效果如图 4.10 所示。

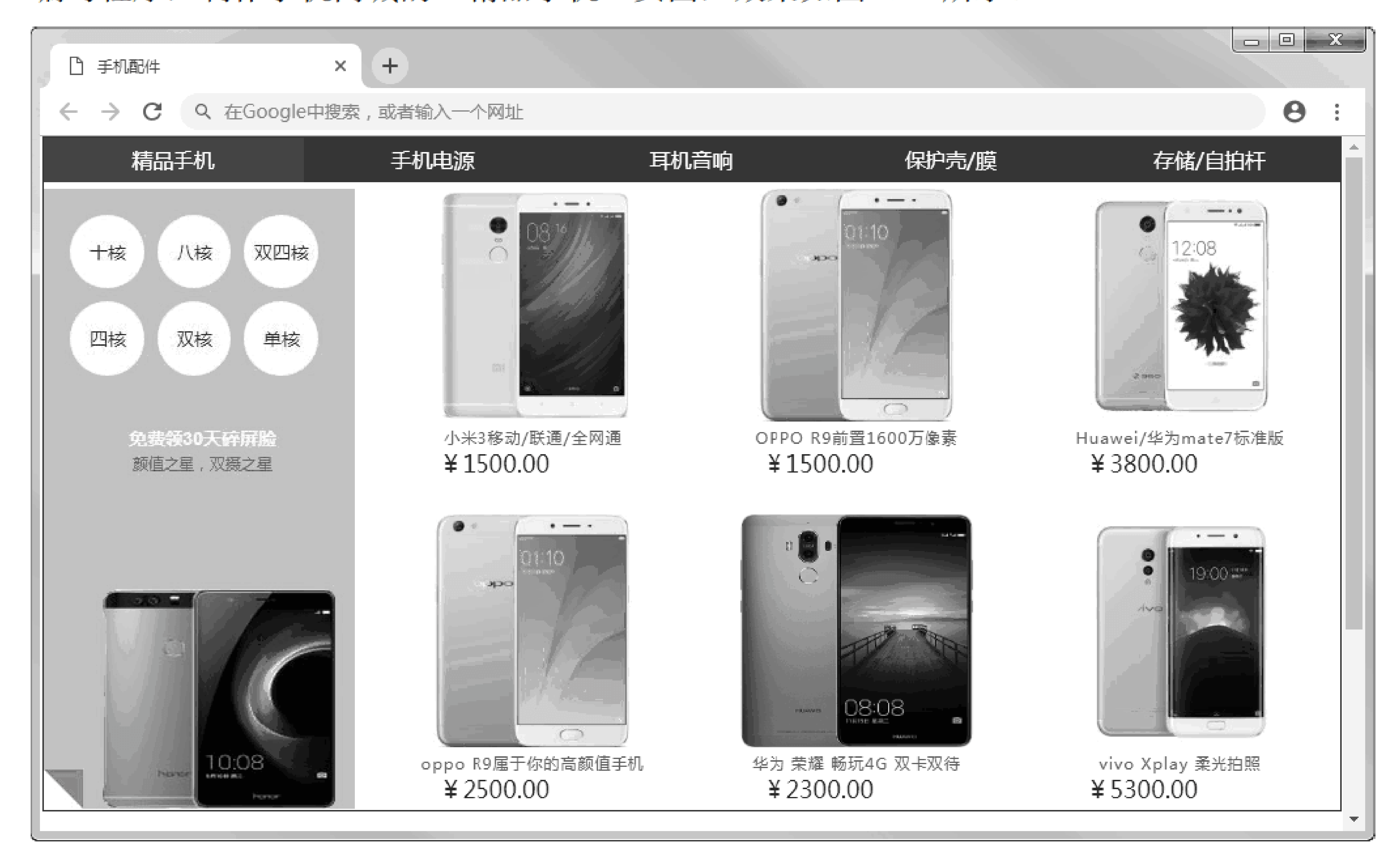


图 4.10　“精品手机”页面

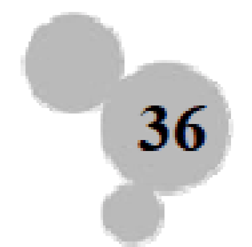
11．制作网站登录页面　　▷①②③④⑤⑥

编写程序，制作 51 购商城的登录页面，效果如图 4.11 所示。

（提示：添加一张背景图片）



图 4.11　网站登录页面



12．制作手机展示页面　　▷①②③④⑤⑥

编写程序，实现手机商城的多款手机展示页面，如图 4.12 所示。
（提示：使用背景图片和文字的相关属性）









图 4.12　手机展示页面

13．华为手机详情页面　　▷①②③④⑤⑥

编写程序，实现华为手机详情页面，如图 4.13 所示。
（提示：使用列表标签实现）

HUAWEI

华为honor荣耀畅玩5A移动版4G大屏手机
¥689

Huawei/华为 P9全网通5.2英寸
¥2199

正品华为/honor荣耀畅玩大屏手机
¥569

【正品】华为/honor荣耀畅玩5A全网通
¥739

华为honor/荣耀畅玩7X全面屏
¥1699

64G高配版Huawei/华为Mate8
¥2028

3GB+32GB全网通徕卡双摄像头
¥2188

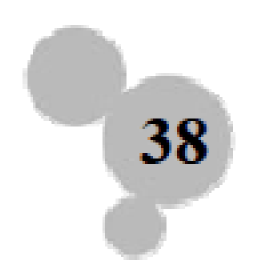
32G全网通华为/Huawei畅玩5C
¥1049.00

图 4.13　商品详情页面

14．手机产品参数页面　▷①②③④⑤⑥

编写程序，制作购物网站中某款手机的产品参数页面，效果如图 4.14 所示。

（提示：使用列表标签实现）



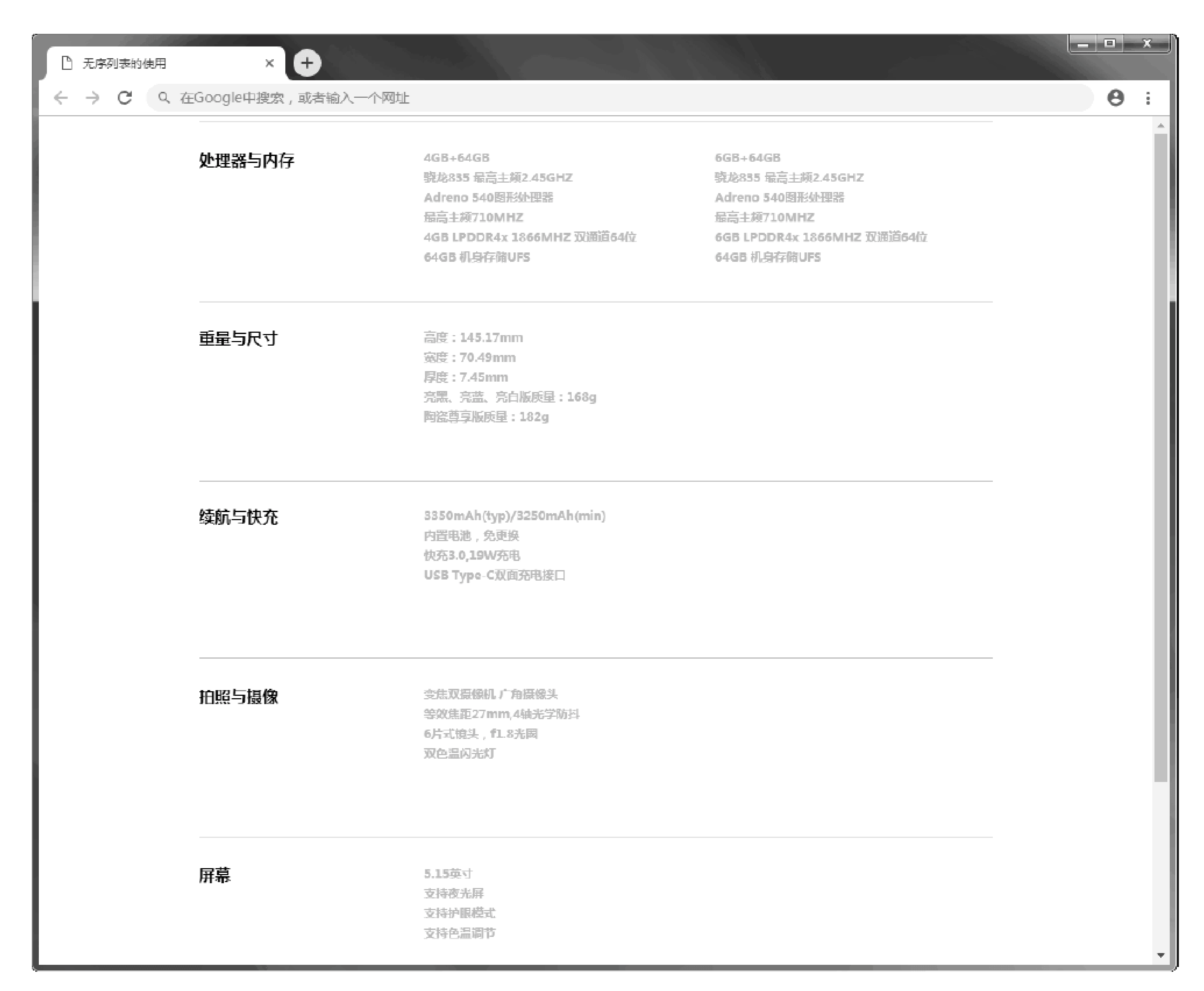


图 4.14　产品参数页面

训练二：实战能力强化训练

15. 图片跑马灯效果　　▷①②③④⑤⑥

以滚动方式显示图片，实现图片跑马灯效果，如图 4.15 所示。页面加载完毕后，图片自动向右滚动；单击向左滚动按钮◁，图片向左滚动；单击向右滚动按钮▷，图片继续向右滚动；鼠标悬停在图片上时，图片停止滚动。



图 4.15　图片跑马灯效果

16．相册内图片单击轮换效果　　▷①②③④⑤⑥

实现相册内图片单击轮换效果。显示 5 张图片，两侧各有一个圆形的半透明按钮，以控制图片向左或向右进行切换。当鼠标指针悬停圆形按钮上时，如悬停在右侧按钮上，该按钮不再透明。单击右侧按钮，图片会逆时针依次移动一个位置；单击左侧按钮，图片可顺时针依次移动一个位置。实现效果如图 4.16 所示，



图 4.16　相册切换效果

17．图形加载动画　　▷①②③④⑤⑥

浏览网页时，难免会遇到页面加载缓慢的情况。为避免网页长时间空白，可以显示一些简单的动画。实现一个圆形加载动画，如图 4.17 所示，页面加载过程中弧形一直转动，直至页面加载完成。

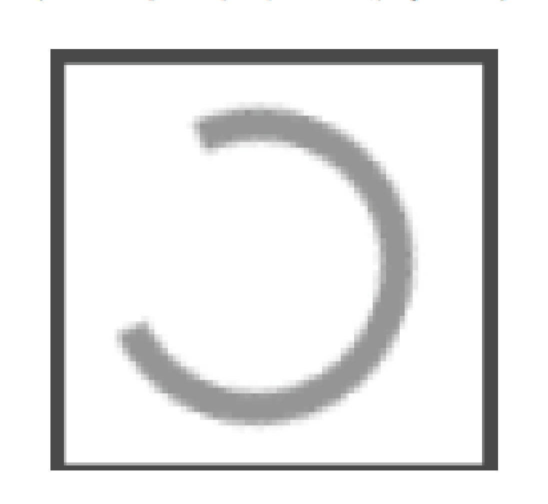
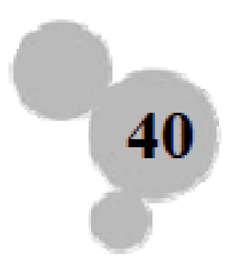


图 4.17　圆形加载动画

18．制作方块填充的进度条　　▷①②③④⑤⑥

下载或安装软件过程中，经常会看到各类进度条，最常见的就是如图 4.18 所示的进度条。试着实现这样的进度条，随着下载或安装进行，绿色方块区逐渐变长，当进度达到 100%时，进度条完全被充满。

图 4.18　方块填充的进度条



第 5 章　CSS3 高级应用



本章训练任务对应核心技术分册第 5 章“CSS3 高级应用”部分。

重点练习内容：

1. 掌握CSS3中的常用的选择器。
2. 使用CSS3设置网页的背景颜色以及背景图像。
3. 熟练掌握盒模型。
4. 灵活使用position、浮动。
5. 使用CSS中的变形、过渡以及动画。

应用技能拓展学习

1. 伪类选择器的应用

浏览网页时，经常会看到一些特殊样式在鼠标悬停、单击、获取焦点时触发。设置这些样式需要使用以下 4 个伪类选择器。

- ☑ :link ： 表示对未访问的超链接应用样式。
- ☑ :visited ： 表示对已访问的超链接应用样式。
- ☑ :hover ： 表示对鼠标悬停的元素应用样式。
- ☑ :active ： 对用户正在单击的元素应用样式。

使用伪类选择器时，需要注意两点：

- ☑ 伪类选择器的顺序必须正确。如果不正确，将无法正确应用样式。正确顺序是:hover 在:link、:visited 之后，:active 在:hover 之后。
- ☑ :link 和:visited 只能用于超链接，不能用于其他元素。

例如，在网页中添加一个导航菜单，并为菜单分别设置未访问、已访问、悬停以及单击时的样式，其关键代码如下：

```
<style type="text/css">
    ul li a:link {                    /*未访问时的超链接样式*/
        color: black;
        text-decoration: none;
    }
```

```
    ul li a:visited {                /*已访问的超链接样式*/
        color: red;
    }
    ul li a:hover {                   /*鼠标悬停时的超链接样式*/
        color: #fff;
        background: #35ccc7;
    }
    ul li a:active {
        text-decoration: underline;   /*正在单击时的超链接样式*/
    }
</style>
<ul>
    <li><a href="../05/index.html">首页</a></li>
    <li><a href="../06/index.html">果蔬生鲜</a></li>
    <li><a href="../07/index.html">户外运动</a></li>
    <li><a href="../08/index.html">手机数码</a></li>
</ul>
```

上述代码的实现效果分别如图 5.1~图 5.4 所示。

首页 果蔬生鲜 户外运动 手机数码

图 5.1　被访问时

首页 果蔬生鲜 户外运动 手机数码

图 5.2　鼠标悬停时

首页 果蔬生鲜 户外运动 手机数码

图 5.3　单击链接时

首页 果蔬生鲜 户外运动 手机数码

图 5.4　访问后

2. 弹性布局（flexbox）的使用

网页布局是 CSS 的一个重点应用，传统的布局依赖于 display+position+float，弹性布局是 CSS3 中一种新的布局方式，使用这种布局的元素被称为弹性容器（flex-box），其所有子元素会自动成为弹性项目（flex-item）。

（1）容器主轴方向

弹性容器存在两根轴——主轴和侧轴。默认情况下，主轴方向为水平向右，侧轴方向为垂直向下。可通过 flex-direction 属性修改主轴方向。

☑ row：默认值，表示主轴方向为水平向右，侧轴方向为垂直向下，如图 5.5 所示。

☑ row-reverse：表示主轴方向为水平向左，侧轴方向为垂直向下，如图 5.6 所示。

☑ column：表示主轴方向为垂直向下，侧轴方向为水平向右，如图 5.7 所示。

☑ column-reverse：表示主轴方向为垂直向上，侧轴方向为水平向右，如图 5.8 所示。



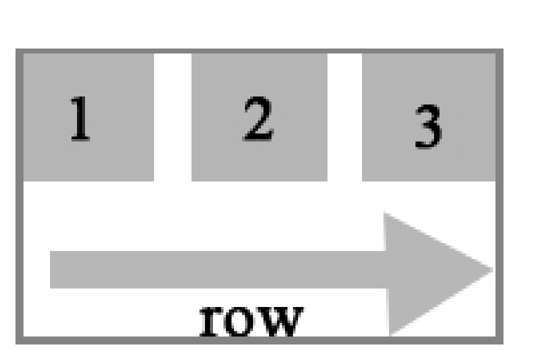


图 5.5　row 排列

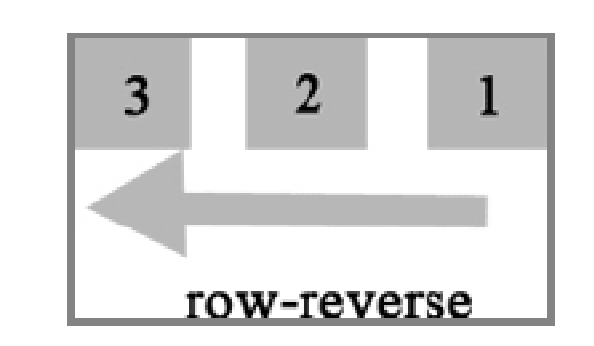


图 5.6　row-reverse 排列

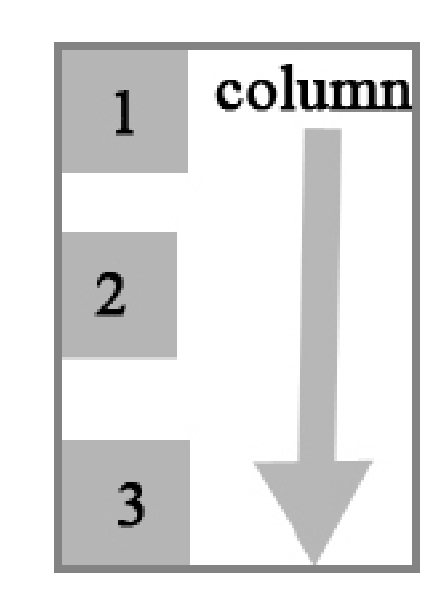


图 5.7　column 排列

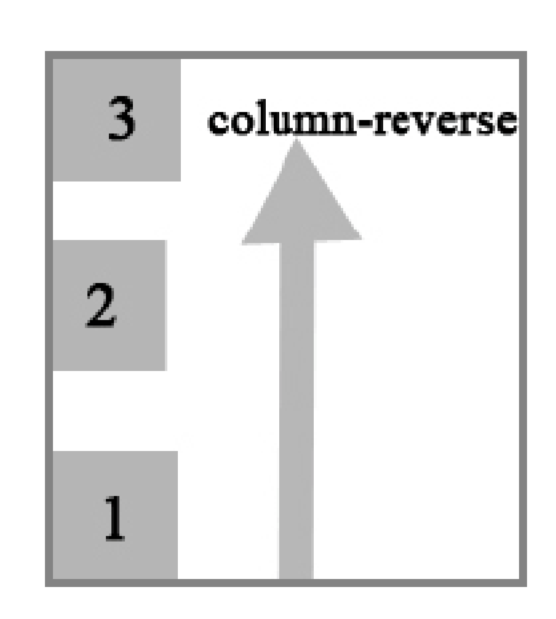


图 5.8　column-reverse 排列

（2）容器属性

除了主轴方向以外，容器还有以下属性。

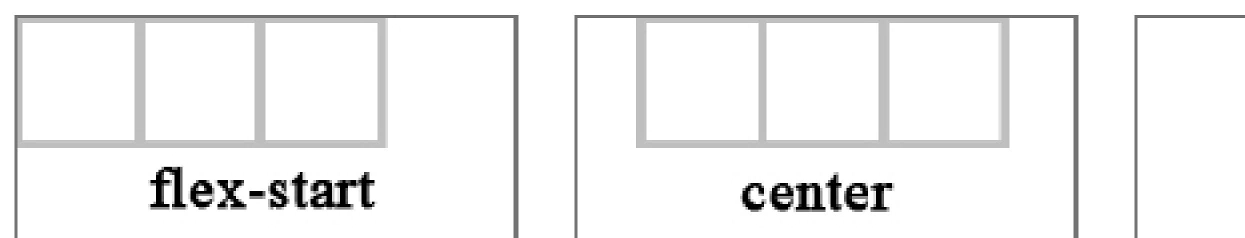
☑ flex-wrap：当项目不能完全显示在一行中时，是否换行以及如何换行，由以下 3 个属值决定。

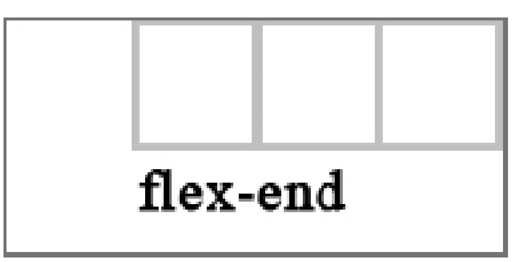
- ➢ no-wrap：项目不换行。
- ➢ wrap：换行，从上往下换行，即第一行在最上方。
- ➢ wrap-reverse：换行，从下往上换行，即第一行在最下方。

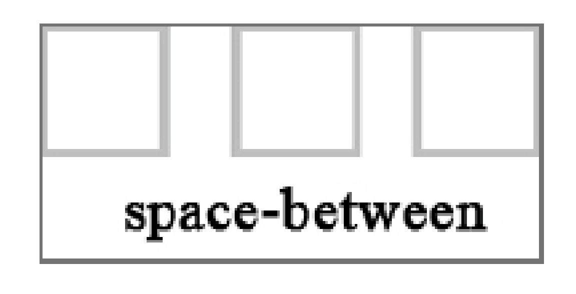
☑ flex-flow：flex-direation 和 flex-wrap 的简写，其语法如下：

```
flex-flow：flex-direction flex-wrap
```

☑ justify-content：设置项目在主轴上的对齐方式。共有 5 个属性值，其值及表示的对齐含义如图 5.9 所示。







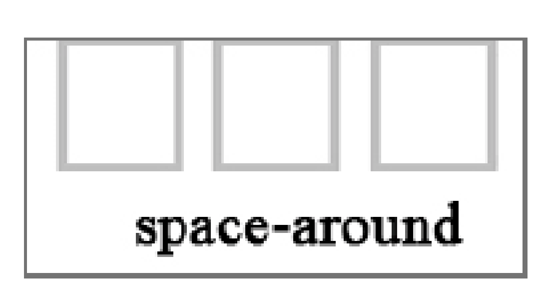
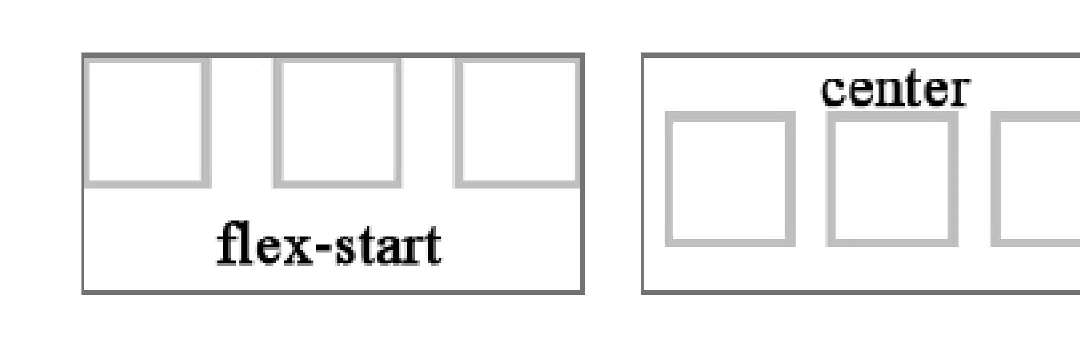
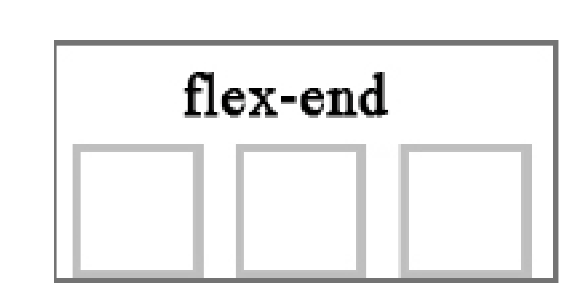
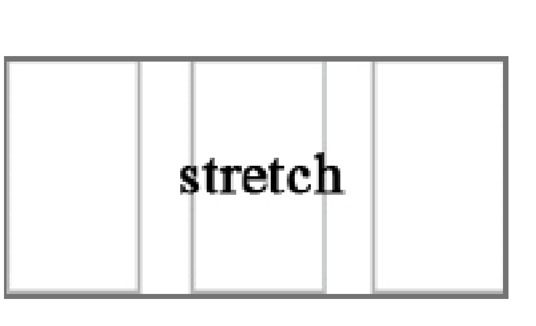


图 5.9　justify-content 属性值

☑ align-items：表示项目在侧轴上的对齐方式。同样有 5 个属性值，其值及表示的对齐含义如图 5.10 所示。







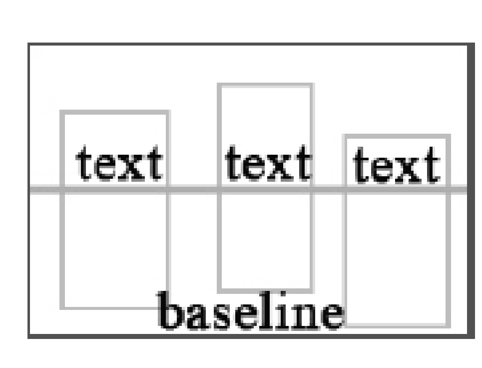
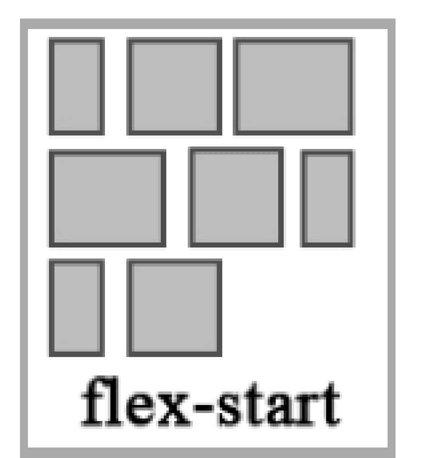
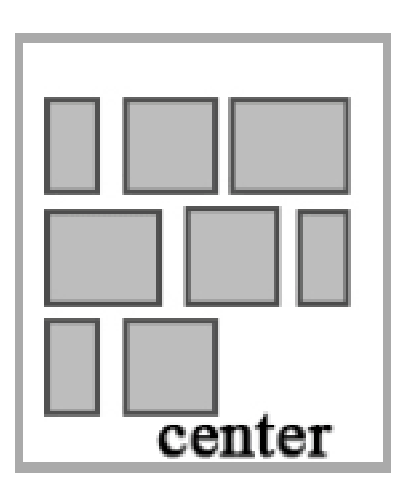
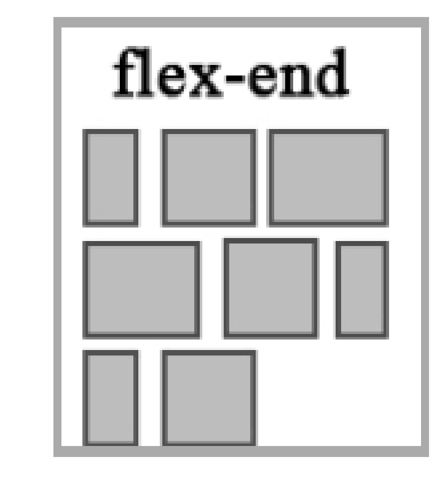


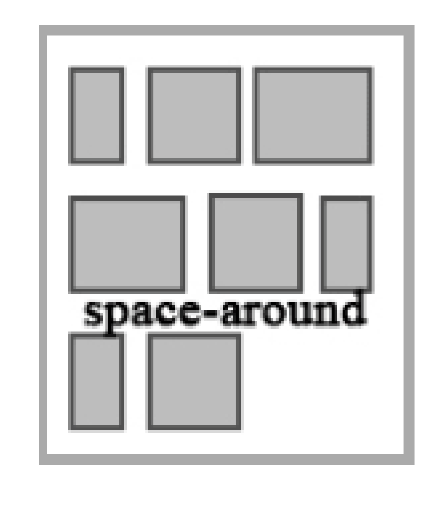
图 5.10　align-items 属性值

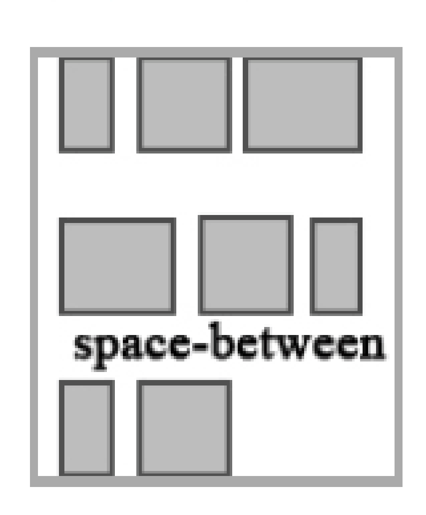
☑ align-content：弹性盒中拥有多行项目时，用于设置行的对齐方式。如果只有一行项目，那么该属性值无效。其主要有 6 个属性值，其值及表示的对齐含义如图 5.11 所示。











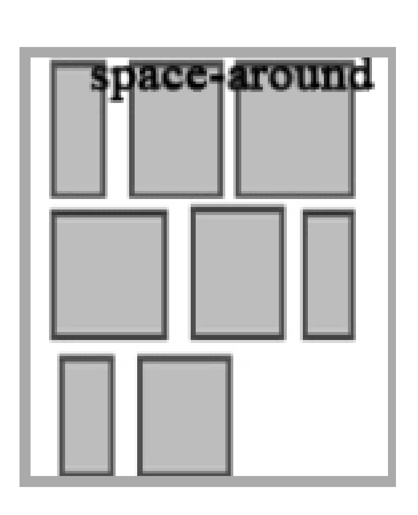


图 5.11　align-content 属性值

（3）项目属性

此外，还可以通过项目属性自定义项目在侧轴上的对齐方式、项目的排列顺序等。项目的具体属

性如下。

- ☑ align-self：自定义某个项目在侧轴上的对齐方式，其属性值及含义与 align-items 相同。
- ☑ flex-grow：定义项目的放大比例。默认值为 0，表示即使有剩余空间，也不放大项目。
- ☑ flex-shrink：定义项目的缩小比例。默认值为 1，表示当空间不足时，缩小项目以适应整个容器。
- ☑ flex-basis：弹性项目分配多余空间前所占的主轴空间。
- ☑ flex：flex-grow、flex-shrink 以及 flex-basis 的缩写。其语法如下：

```
flex：flex-grow flex-shrink flex-basis
```

- ☑ order：定义项目的排列顺序，其属性值为具体数字。数字越小，排列越靠前。

实战技能强化训练

训练一：基本功强化训练

1. 制作手机商城的“精品配件”页面 ▷①②③④⑤⑥

网购是非常受年轻人青睐的一种购物方式。如图 5.12 所示为手机商城的“精品配件”页面，使用 HTML 和 CSS 完成该页面。



图 5.12 “精品配件”页面

2．制作开心消消乐网站的“最新活动”页面　▷①②③④⑤⑥

开心消消乐是一款老少皆宜的休闲游戏。尝试制作游戏网站中的“最新活动”页面，如图 5.13 所示。

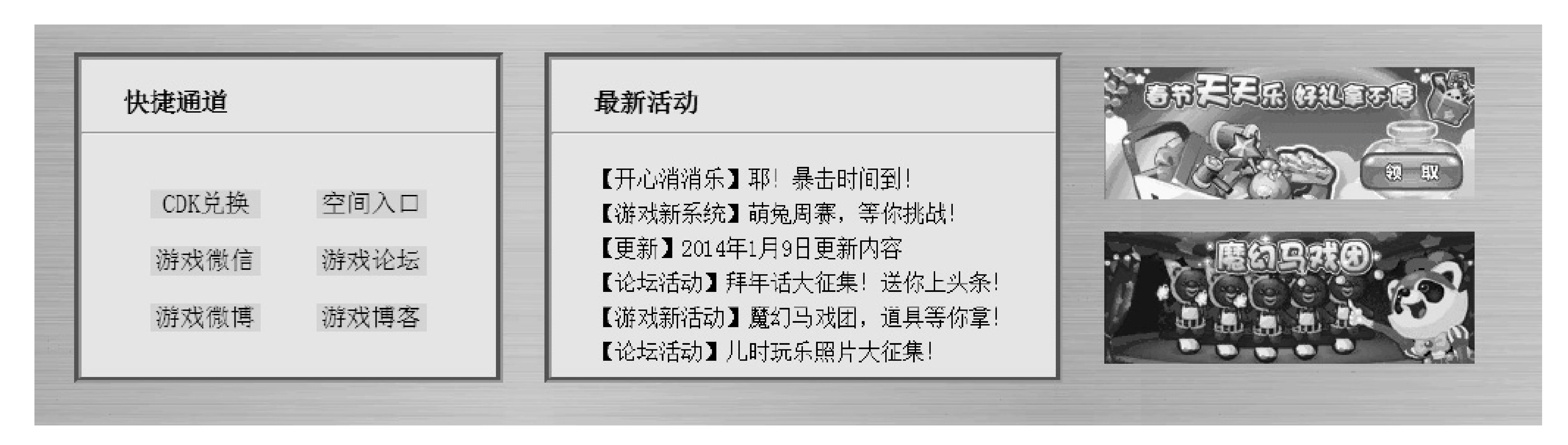


图 5.13　开心消消乐网站的“最新活动”页面

3．制作小米 Max 2 手机宣传页面　▷①②③④⑤⑥

手机的更新速度越来越快，如图 5.14 所示为小米 Max 2 上市时宣传页面的一部分，试着完成这个页面。



图 5.14　小米 Max 2 手机宣传页面

4．实现手机分类筛选页面　▷①②③④⑤⑥

电商网站，为了方便用户能快速找到自己所需的东西，其商品列表页面中通常会设置一些条件筛

选项。如图 5.15 所示为一个手机页面的分类筛选项，请实现该页面。

品牌	oppo　mi小米　HUAWEI　honor　vivo
分类	手机
网络类型	移动　联通　电信　全网通
机身内存RAM	16G　32G　64G　128G

图 5.15　手机分类筛选项

5．制作新增收货地址页面　▷①②③④⑤⑥

网购中，选择好商品后，需要选择或添加收货地址，为下一步支付做准备。如图 5.16 所示就是一个添加收货地址的页面，试着实现这个页面。

新增收货地址

所在地区：

详细地址：

邮政编码：

收货人姓名：

收货人电话：

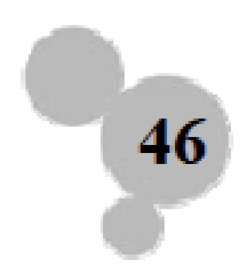
是否允许代收：○是　○否

保存

图 5.16　新增收货地址页面

6．制作商品详情页的产品规格部分　▷①②③④⑤⑥

编写程序，制作商品详情页的产品参数与规格部分，效果如图 5.17 所示。



产品参数

图案：卡通动漫	颜色：人物印花	袖型：中袖
上衣厚度：适中	上衣门襟：单排扣	穿着方式：开衫
尺码：S，M	衣长：30-40cm	版型：宽松
下摆：敞口	季节：夏季	材质：其他
领型：圆领	潮流：日系	风格：通勤（OL）

图 5.17　商品详情页的产品参数部分

7．制作手机商城中“主题购”页面　▷①②③④⑤⑥

如图 5.18 所示为电子商城中的“主题购”页面，尝试完成这个页面。



图 5.18　“主题购”页面

8．制作一个简单的横向导航栏　▷①②③④⑤⑥

编写一个简单的横向导航栏，当鼠标指针悬停在某个菜单项上时，该菜单项会自动添加背景颜色，效果如图 5.19 所示。

（提示：使用背景以及文本相关属性）



图 5.19　简单导航栏

9. 制作含二级菜单的横向导航栏　▷①②③④⑤⑥

当导航内容较多时，为了节省空间，通常会设置多级导航菜单，将一部分内容折叠起来。如图 5.20 所示为一个横向二级导航菜单页面，当鼠标指针悬停在一级菜单项上时，会自动展开二级子菜单项，请编程实现该页面。



图 5.20　横向导航栏

10. 制作含二级菜单的侧边导航栏　▷①②③④⑤⑥

侧边导航也是一种比较常见的导航方式，其导航菜单是竖向的，子菜单也同步向右展开。如图 5.21 所示为某电商网站中当鼠标悬停在第 5 项菜单上时展开的子菜单效果。请编程实现这个页面。

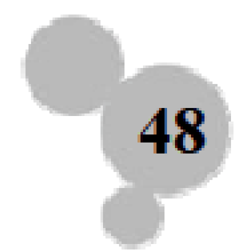




图 5.21　含二级导航菜单的侧边导航栏

11. 旋转风车效果　▷①②③④⑤⑥

制作一个旋转风车页面，当鼠标指针悬停在风车上时，风车开始旋转，效果如图 5.22 所示。（提示：旋转效果中，通过 transform-orgin 属性为旋转对象设置旋转中心点）



图 5.22　旋转风车效果

12．自动拼图动画效果 ▷①②③④⑤⑥

编写程序，实现自动拼图的动画效果。打开页面时，若干图片散乱地显示在页面中，如图 5.23 所示。当鼠标放置在任意一张图片上时，所有图片会自动组合为一张完整图像，如图 5.24 所示。

（提示：通过定位 position 以及旋转等效果实现图像拼接前的位置，通过 CSS3 中的过渡属性实现图像自动拼接过程）



图 5.23　图片散乱显示



图 5.24　拼图效果

13．鼠标光标滑过时的平移特效 ▷①②③④⑤⑥

编写程序，实现电子商城中当鼠标光标滑过商品时，商品逐次平移的页面效果，如图 5.25 所示。



图 5.25　鼠标光标滑过时的平移特效

14．鼠标悬停时展开和放大相册 ▷①②③④⑤⑥

编写程序，制作一个网页版相册。当鼠标悬停在相册上时，相册展开；鼠标再次悬停在照片上时，照片逐渐放大，如图 5.26 所示。

（提示：通过伪类选择器可以实现鼠标悬停特效）





图 5.26　网页版相册

15. 旅游网站图片轮播效果　▷①②③④⑤⑥

编写程序，实现图片依次向左轮流播放的页面效果，如图 5.27 所示。

（提示：通过 animation 实现轮播图动画）

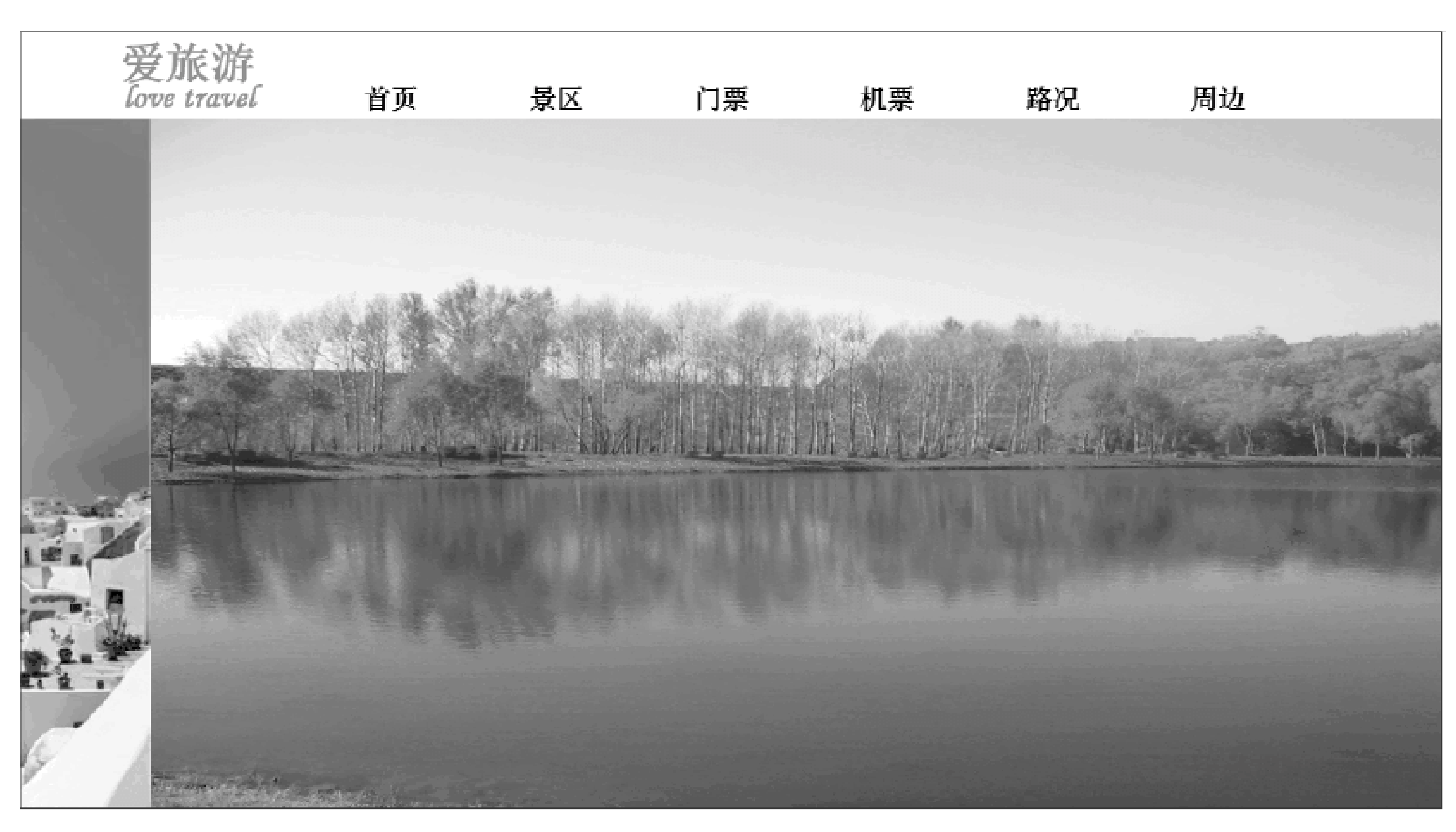


图 5.27　图片轮播效果

16. 广告页面文字滚动显示效果　▷①②③④⑤⑥

编写程序，实现网站中文字滚动显示的页面效果，如图 5.28 所示。

（提示：通过 animation-dalay 属性设置动画延长时间，使得广告按顺序滚动显示）



图 5.28　广告中文字滚动显示效果

训练二：实战能力强化训练

17．实现凸显图片效果　▷①②③④⑤⑥

实现一个画廊图片效果，如图 5.29 所示。画廊中的图片整齐排列，却又很模糊，当鼠标悬停在某张图片上时，该图片会逐渐变清晰。

（提示：调整图片清晰度，可使用 CSS3 中的 opacity 属性实现，其值为 0～1 的小数）

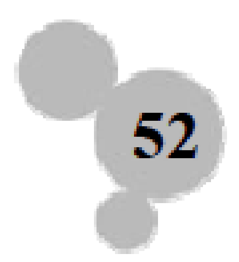


图 5.29　画廊图片效果

18．实现文字水波纹效果　▷①②③④⑤⑥

浏览网页时，常会看到各种各样的文字特效。试着通过多个动画组合，实现如图 5.30 所示的文字水波纹效果。

（提示：使用 CSS 的 background-clip 属性实现）



清水出芙蓉，天然去雕饰

图 5.30　文字水波纹效果

19．实现邮件订阅中心页面　▷①②③④⑤⑥

每天打开手机，看看娱乐新闻或推送的文章，已成为多数年轻人的习惯。有的人喜欢订阅一些邮件，每早收到一些感兴趣的推荐和资讯。如图 5.31 所示就是通过弹性布局实现的邮件订阅中心页面，当页面内容缩小到无法装下一整行内容时，页面中的项目就会分两行排列，如图 5.32 所示，请尝试实现该页面。



图 5.31　邮件订阅中心



图 5.32　分两行显示

20．制作红包兑换页面　▷①②③④⑤⑥

在淘宝、美团等 APP 上签到或消费后，经常会有发红包、领红包活动，参与即可领取一些优惠券。试着用弹性布局制作一个兑换红包页面，效果如图 5.33 所示。

21．制作 360 每日趣闻页面　▷①②③④⑤⑥

安装过 360 的人都知道，每天一打开电脑，就会弹出一个推送页面，涵盖脑筋急转弯、最新资讯、游戏等信息。如图 5.34 所示就是通过弹性布局实现的一个仿 360 的每日趣闻页面，尝试完成这个页面。



图 5.33　红包兑换页面



图 5.34　360 每日趣闻页面

22．实现跳动文字效果

▷①②③④⑤⑥

编写程序，实现文字在页面中上下跳动的动画，效果如图 5.35 所示。

mingrisoft

图 5.35　跳动的文字

第 6 章　表格与<div>标签



本训练任务对应核心技术分册第 6 章“表格与<div>标签”部分。

重点练习内容：

1. 表格的基本使用。
2. 表格中合并单元格的使用。
3. 表格标签中各子标签及其属性的含义。
4. <div>的含义及其使用。
5. DIV+CSS的结合使用。

应用技能拓展学习

1. 表格中的结构标签

使用<thead> <tbody>和<tfoot> 3 组标签，可将表格分为表头、表文内容和页脚 3 部分。这 3 组标签需要结合使用，通过使用这些标签可以滚动表格的主题内容。

例如，在网页中添加一个大学课程表，关键代码如下：

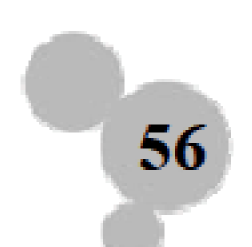
```
<table border="1" cellpadding="0" cellspacing="0" width="800" bgcolor="#FFD7FE" height="300">
  <!—表头-->
    <thead>
    <tr bgcolor="#FFB039">
        <th></th>
        <th>星期一</th>
        <th>星期二</th>
        <th>星期三</th>
        <th>星期四</th>
        <th>星期五</th>
    </tr>
    </thead>
  <!—表格主题-->
    <tbody>
```

```
    <tr>
        <th>08:00-09:30</th>
        <td>大学语文</td>
        <td>计算机基础理论</td>
        <td>大学英语</td>
        <td></td>
        <td>计算机算法</td>
    </tr>
    <tr>
        <th>09:50-11:20</th>
        <td>计算机实践</td>
        <td>逻辑算法</td>
        <td></td>
        <td>C 语言基础</td>
        <td></td>
    </tr>
    <!--此处省略其余内容-->
    </tbody>
  <!—页脚-->
    <tfoot>
    <tr bgcolor="#FFB039">
        <td colspan="6" align="center">备注：星期五下午，全年级进行大扫除</td>
    </tr>
    </tfoot>
</table>
```

上述代码的实现效果如图 6.1 所示。

	星期一	星期二	星期三	星期四	星期五
08:00-09:30	大学语文	计算机基础理论	大学英语		计算机算法
09:50-11:20	计算机实践	逻辑算法		C语言基础	
12:50-14:20		大学英语			计算机基础理论
14:40-16:10	多媒体技术		大学物理		
16:30-18:00			商务英语（选修）		
备注：星期五下午，全年级进行大扫除					

图 6.1　在网页中添加课程表



2. 伪元素选择器的巧用

伪元素选择器可用来改变文档中特定部分的效果样式，这是通过普通选择器无法定义到的部分。CSS 中常用的伪元素选择器有以下 4 种。

- ☑ :first-letter：指定对象中第一个字符的样式。
- ☑ :first-line：指定对象内第一行文字的样式。
- ☑ :before：在指定对象的前面添加内容。
- ☑ :after：在指定对象的后面添加内容。

在使用:after 和:before 时，需要先通过 content 属性设置要添加的内容，否则设置的样式无效。例如训练 9 中，邀请函中的菱形方块就是通过:after 实现的。

实战技能强化训练

训练一：基本功强化训练

1. 表格实现键盘快捷键介绍页面　▷①②③④⑤⑥

使用电脑办公时，要想提高工作效率，掌握一些常用的快捷键是必不可少的。试着编程实现一个网页，显示常见的快捷键组合及其功能，如图 6.2 所示。

（提示：设置表格隔行换色时，可通过:nth-child(2n)和:nth-(2n+1)分别选中表格中的偶数行和奇数行）

功能	Windows	Mac OS
编辑菜单	Alt+E	Ctrl+F2+F
文件菜单	Alt+F	Ctrl+F2+E
视图菜单	Alt+V	Ctrl+F2+V
全选	Ctrl+A	Cmd+A
复制	Ctrl+C	Cmd+C
查找文本	Ctrl+F	Cmd+F
新建文档	Ctrl+N	Cmd+N
打开文件	Ctrl+O	Cmd+O
剪切文本	Ctrl+X	Cmd+X
撤销文本	Ctrl+Z	Cmd+Z
打印选项	Ctrl+P	Ctrl+P

图 6.2　表格实现键盘快捷键介绍

2. 表格实现 12306 公告页面　▷①②③④⑤⑥

表格曾经是流行一时的页面布局方式，虽然现在人们更多使用 DIV+CSS 布局页面，但在布局一些简单页面时仍然可以使用表格来实现。使用表格，布局如图 6.3 所示的 12306 网站公告部分。

最新发布		常见问题	信用信息
候补的购票操作说明	2019-07-23	公告候补的购票操作说明	2018-11-03
关于调整互联网、电话订票起售时间…	2019-07-27	中国铁路广州局有限公司关于2019年8月…	2019-08-29
候补的购票操作说明	2019-07-23	公告候补的购票操作说明	2018-11-03
关于调整互联网、电话订票起售时间…	2019-07-27	中国铁路广州局有限公司关于2019年8月…	2019-08-29
候补的购票操作说明	2019-07-23	公告候补的购票操作说明	2018-11-03
关于调整互联网、电话订票起售时间…	2019-07-27	中国铁路广州局有限公司关于2019年8月…	2019-08-29

图 6.3　表格实现 12306 公告页面

3. CSS 实现唯美主题背景　▷①②③④⑤⑥

每个网站都有自己的风格，如图 6.4 所示为使用 CSS 实现的明日科技唯美风格页面效果，试着编程实现该页面效果。

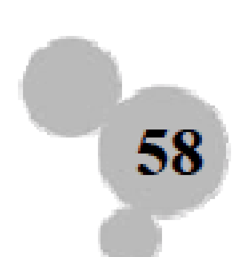


图 6.4　明日科技唯美主题页面

4. 表格实现 KTV 价格表　▷①②③④⑤⑥

饭后飙歌是年轻人聚会时常见的娱乐方式。如图 6.5 所示为印象 KTV 某分店的价格表，试着编程实现该页面。

（提示：表头斜线可通过伪元素选择器实现）



时段 / 每小时	小包	中包	大包	豪华大包
09:00-18:00(白场)	80	100	200	500
22:00-07:00(午夜场)	100	200	300	800
温馨提示：需一次性消费对应时间段金额可减免房费				
地址：**领域3楼				

图 6.5　印象 KTV 价格表

5．实现电子版违章罚单　▷①②③④⑤⑥

有车族可以在网上查询自己是否有交通违章行为。如图 6.6 所示为电子版的违章罚单效果，试着编程实现该页面。

6．实现 ATM 机银行凭证　▷①②③④⑤⑥

编写一个程序，实现 ATM 机打印的银行凭证效果，如图 6.7 所示。



图 6.6　电子版违章罚单

**银行

客户通知书　　CUSTOMER ADVICE

卡号 CARD NO.			
金额 AMOUNT			受理行号 ACQUIRER NUMBER
转入账号 TRANSFER TO			
存款金额 DEPOSIT			应答码 RESPONSE NUMBER
存款 DEPOSIT	取款 WITHDRAWAL	转账 TRANSFER	其他 OTHER TRANSACTION
接受 ACCEPTED	取消 REJECTED	吞卡 ACRDRETAINED	请与银行联系 PLEASE CONTACT BANK
日期和时间 DATE & TIME			ATM编号 ATM ID
检索参考号 TRANS REFER NO.	检索参考号 RETRIEVAL REFERENCE NUMBER		

如有疑问请持此凭条与银行联系
Please contact the bank with this advices for any questions
客户服务热线电话 9****。 Tel：9****

图 6.7　ATM 机凭证

7．实现健身房课程表

健身馆每天会安排不同的课程，人们可根据课程表合理安排自己的健身训练时间。

如图 6.8 所示为某健身馆的课程表，同一课程的背景颜色相同。请尝试完成这个页面。

（提示：设置单元格背景颜色时，可在单元格中依次通过 bg 属性设置，也可以添加类名，然后使用 CSS 来设置）

<table>
<tr><th></th><th>周一</th><th>周二</th><th>周三</th><th>周四</th><th>周五</th><th>周六</th><th>周日</th></tr>
<tr><td>11:50-12:50</td><td>普拉提</td><td>瑜伽</td><td>哈拉瑜伽</td><td>普拉提</td><td>瑜伽</td><td>哈拉瑜伽</td><td>普拉提</td></tr>
<tr><td>16:30-17:00</td><td colspan="4">哈拉瑜伽</td><td>普拉提</td><td colspan="2">哈拉瑜伽</td></tr>
<tr><td>12:50-13:50</td><td colspan="2"></td><td>肚皮舞</td><td></td><td>肚皮舞</td><td></td><td>哈拉瑜伽</td></tr>
<tr><td>17:30-16:20</td><td colspan="2">有氧舞蹈</td><td colspan="2">肚皮舞</td><td>有氧舞蹈</td><td>肚皮舞</td><td>瑜伽</td></tr>
<tr><td>16:30-17:20</td><td colspan="7">动感单车</td></tr>
<tr><td>17:30-20:15</td><td></td><td colspan="3">动感单车</td><td colspan="3"></td></tr>
</table>

图 6.8　健身房课程表

8．制作电子邀请函　▷①②③④⑤⑥

每逢重大活动，总要邀请各类嘉宾，因此总要设计出各种精美的邀请函。如图 6.9 所示为使用 DIV+CSS 实现的一封简易电子邀请函，试着实现该页面。



图 6.9　邀请函

9．DIV 实现岗位招聘页面　▷①②③④⑤⑥

每逢毕业季，很多毕业生就会收到各公司的招聘宣传单。试着实现一个简易的电子招聘宣传单，其页面效果如图 6.10 所示。

10．表格实现手机版天气预报　▷①②③④⑤⑥

编写一个程序，实现手机端的天气预报页面，效果如图 6.11 所示。

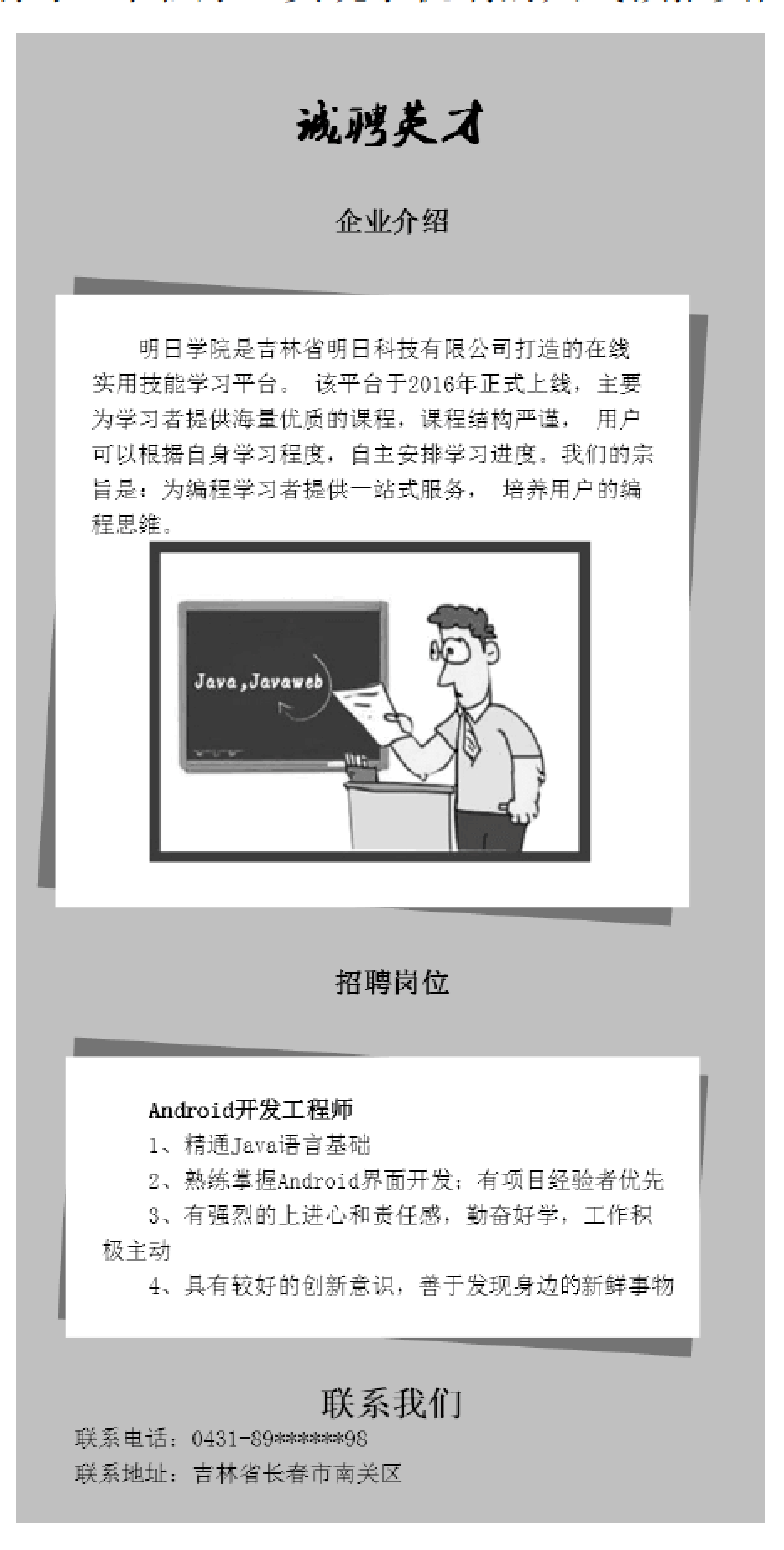


图 6.10　岗位招聘页面

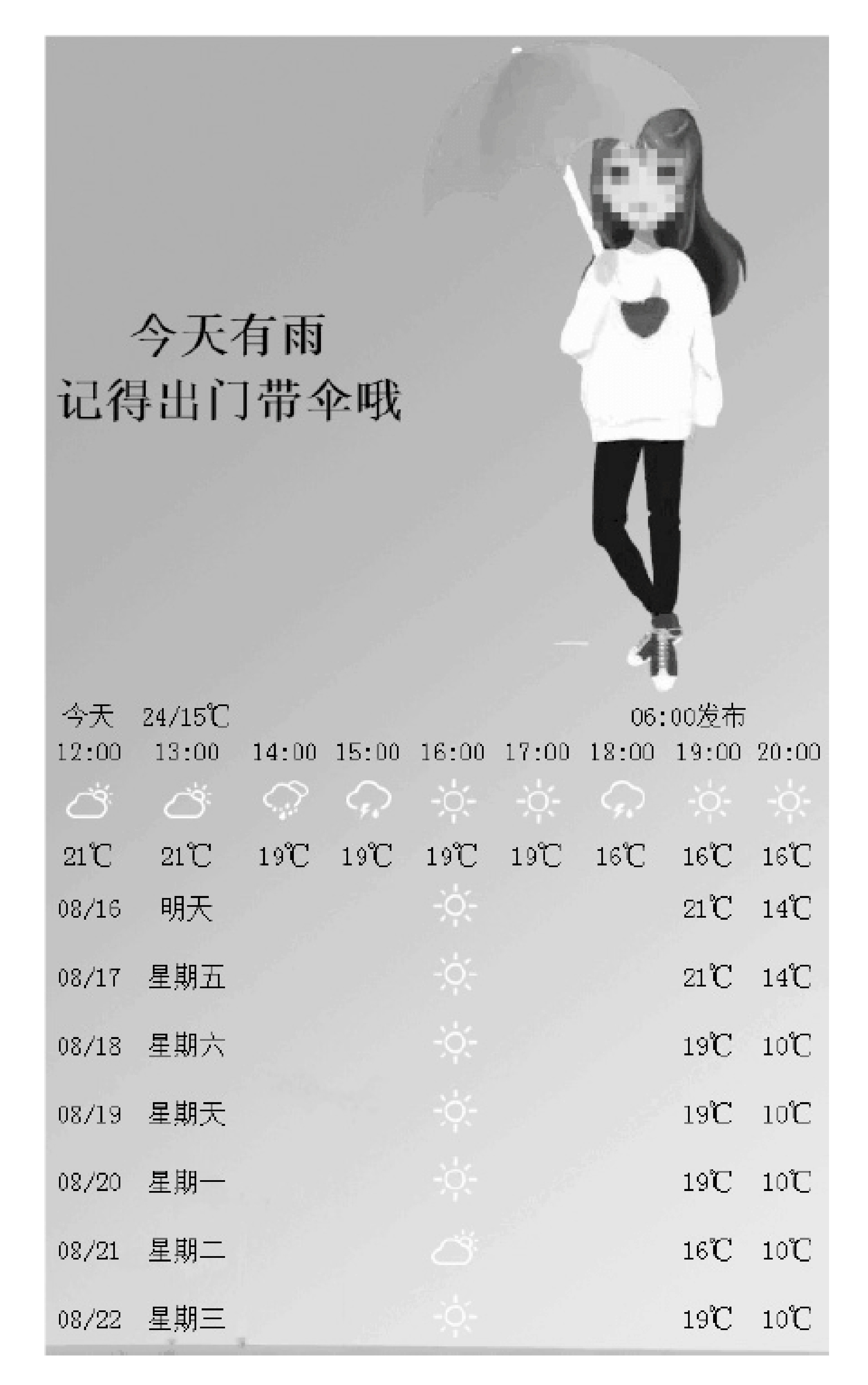


图 6.11　手机版天气预报

11．表格实现商品列表　▷①②③④⑤⑥

布局商品列表页面时通常使用 DIV+CSS 实现，其实表格也可以实现商品列表页面。试着编程实现如图 6.12 所示的水果推荐购买列表页面，要求采用表格来实现。



图 6.12　水果推荐购买列表

12．表格实现 51 购商城首页商品信息　▷①②③④⑤⑥

对于电商网站，相信大家都不陌生。如图 6.13 所示为 51 购商城的首页商品信息，请使用表格实现该页面。



图 6.13　51 购商城首页商品信息

13．制作移动端王者荣耀官网页面　▷①②③④⑤⑥

王者荣耀是一款广受年轻人青睐的游戏。编写程序，实现移动端王者荣耀游戏页面，效果如图 6.14 所示。

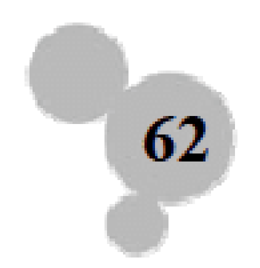




图 6.14 王者荣耀官网页面

14．制作美团外卖页面 ▷①②③④⑤⑥

外卖是近些年来兴起的一个行业，深受广大上班族的喜爱。如图 6.15 所示为美团订餐页面，试着编程实现该页面。



图 6.15 美团外卖页面

15. 表格实现科学计算器 ▷①②③④⑤⑥

编写程序，实现一个简易的科学计算器页面，其实现效果如图 6.16 所示。

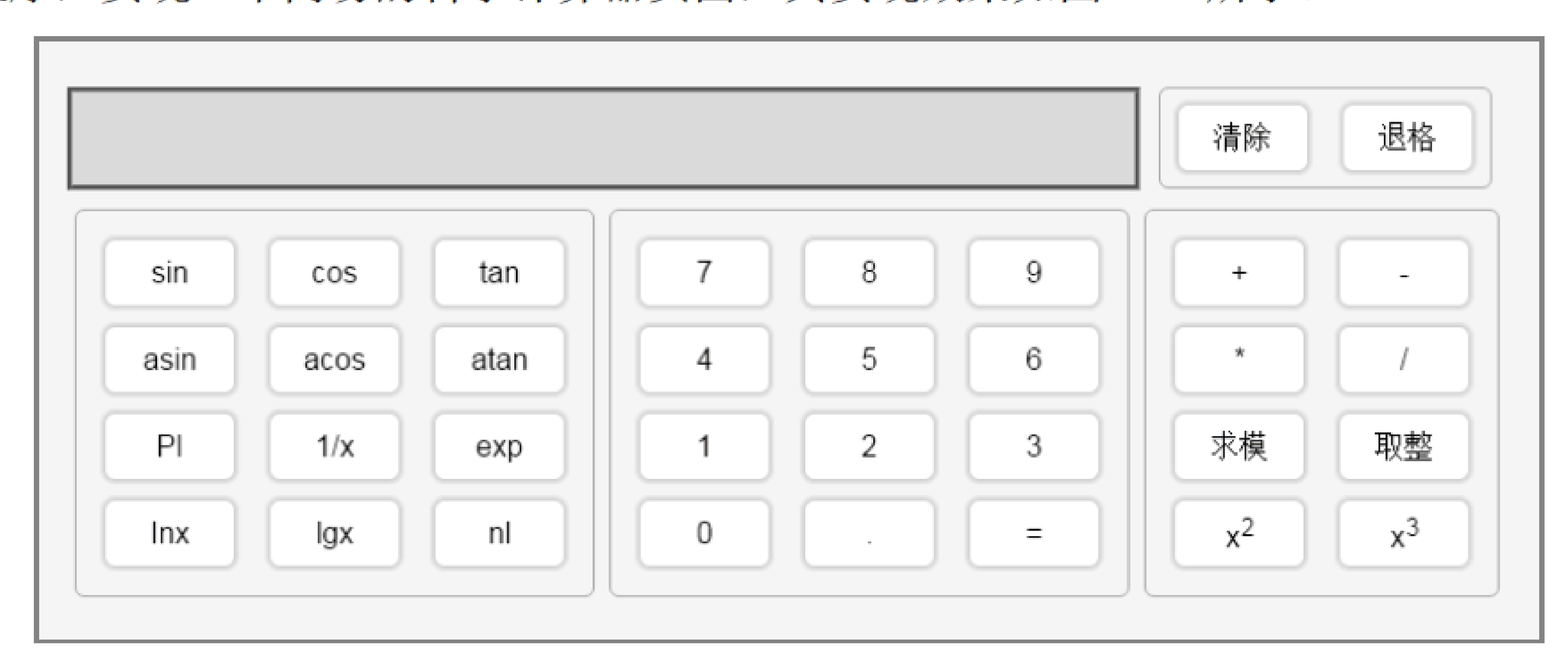


图 6.16 表格实现计算器页面

16. 表格实现日历页面 ▷①②③④⑤⑥

编写程序，实现一则日历页面效果。如图 6.17 所示。

（提示：设置单元格的边框不重复显示，可以通过 CSS 去掉一部分单元格边框）

2019JUN

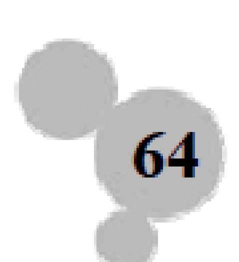
Su	M	Tu	W	Th	F	Sa
						1 儿童节
2 廿九	3 五月大	4 初二	5 初三	6 爱眼日	7 端午节	8 初六
9 初七	10 初八	11 初九	12 初十	13 十一	14 十二	15 十三
16 父亲节	17 十五	18 十六	19 十七	20 十八	21 夏至	22 二十
23 廿一	24 廿二	25 廿三	26 廿四	27 廿五	28 廿六	29 廿七
30 廿八						

图 6.17 表格实现日历页面

17. DIV 实现优惠券领取页面 ▷①②③④⑤⑥

打开美团或饿了么 App 订餐时，总会弹出优惠券领取页面，如图 6.18 所示。试着编写程序实现该页面效果。

（提示：调整图片清晰度可使用 CSS3 中的 opacity 属性实现，其属性值为 0～1 的小数）



18．微信朋友圈动态效果　　▷①②③④⑤⑥

刷朋友圈几乎是当代人每天做的一件必不可少的事。试着编写程序，实现仿微信朋友圈的文章列表效果，如图 6.19 所示。



图 6.18　优惠券领取页面



图 6.19　微信朋友圈页面

训练二：实战能力强化训练

19．图文显示课程列表页面　　▷①②③④⑤⑥

如图 6.20 所示为通过弹性布局实现的课程列表页面，试着编程实现这个页面。

20．使用表格布局女装页面　　▷①②③④⑤⑥

如图 6.21 为使用表格实现的精美女装页面，试着编程实现该页面。

21．美食制作热门推荐列表

▷①②③④⑤⑥

如图 6.22 所示为某美食视频网站的热门推荐页面。当鼠标指针悬停某个视频教程上时，视频下方会向上滑出一个播放按钮。试着编程实现该页面。



图 6.20　课程列表页面



图 6.21　使用表格布局女装页面





图 6.22　美食制作视频教程页面

22．表格实现游戏博客页面　▷①②③④⑤⑥

编写一个程序，利用表格实现一个游戏博客页面，具体效果如图 6.23 所示。



图 6.23　游戏博客页面



第 7 章　列　　表

本章训练任务对应核心技术分册第 7 章“列表”部分。

重点练习内容：

1. 3类常用列表的基本使用。
2. 使用CSS3对列表项进行横向和纵向排列。
3. 列表项样式的设置。
4. 列表的嵌套。

应用技能拓展学习

1. CSS3 中的列表属性

使用无序列表或有序列表时，使用 type 属性可设置列表项的标志或序号类型。CSS3 中 list-style 属性可用来定义列表项的标志和位置。其语法格式如下：

```
list-style:list-style-type list-style-position list-style-image
```

☑ list-style-type：设置列表项标志的类型，其值有 circle（空心圆）、square（实心方块）等，默认值为 disc（实心圆）。

☑ list-style-position：表示列表项标志的位置，其值有 inside（位于文本内，环绕文本根据标记对齐）和 outside（位于文本左侧，放置在文本外）以及 inherit（从父元素继承 list-style-position 属性的值）。

☑ list-style-image：将图像作为列表项的标志，其值为图像的路径。

（提示：list-style 是一个简写属性，实际使用时既可以直接使用该属性，也可以单独使用其他列表属性。）

2. 列表的嵌套

有序列表、无序列表以及定义列表之间可以互相嵌套。嵌套的方式不一样，表现结果也不一样。实际开发时，可根据需要选择适当的嵌套形式。

例如，通过嵌套有序列表，实现一个刮刮乐介绍页面，代码如下：

```
<style type="text/css">
    dl {                    /*设置背景*/
        background: url('bg.jpg') no-repeat;
        background-size: 100% 100%;
        width: 600px;
        padding: 180px 0 50px
    }
    dt {                    /*标题*/
        font-size: 40px;
        width: 250px;
        margin: 0 auto;
        border-radius: 10px;
        color: #af4040;
    }
    ol li {                              /*有序列表*/
        margin: 16px 0px;
        font :bold   22px/40px "";
        color: #272220;;
        list-style-position: inside;
        width: 480px;
        background: rgba(255, 194, 142, 0.3);
        border-radius: 10px;
    }
    </style>
<dl>
    <dt style="text-align: center;"><span style="font-size: 46px">刮刮乐</span>常识</dt>
    <dd style="margin:180px 0 0 20px">
        <ol>
            <li>什么是刮刮乐</li>
            <li>刮刮乐生产过程大揭秘</li>
            <li>刮刮乐购票须知</li>
            <li>刮刮乐常见玩法汇总</li>
            <li>刮刮乐十大主题代表游戏</li>
        </ol>
    </dd>
</dl>
```

其运行效果如图 7.1 所示。

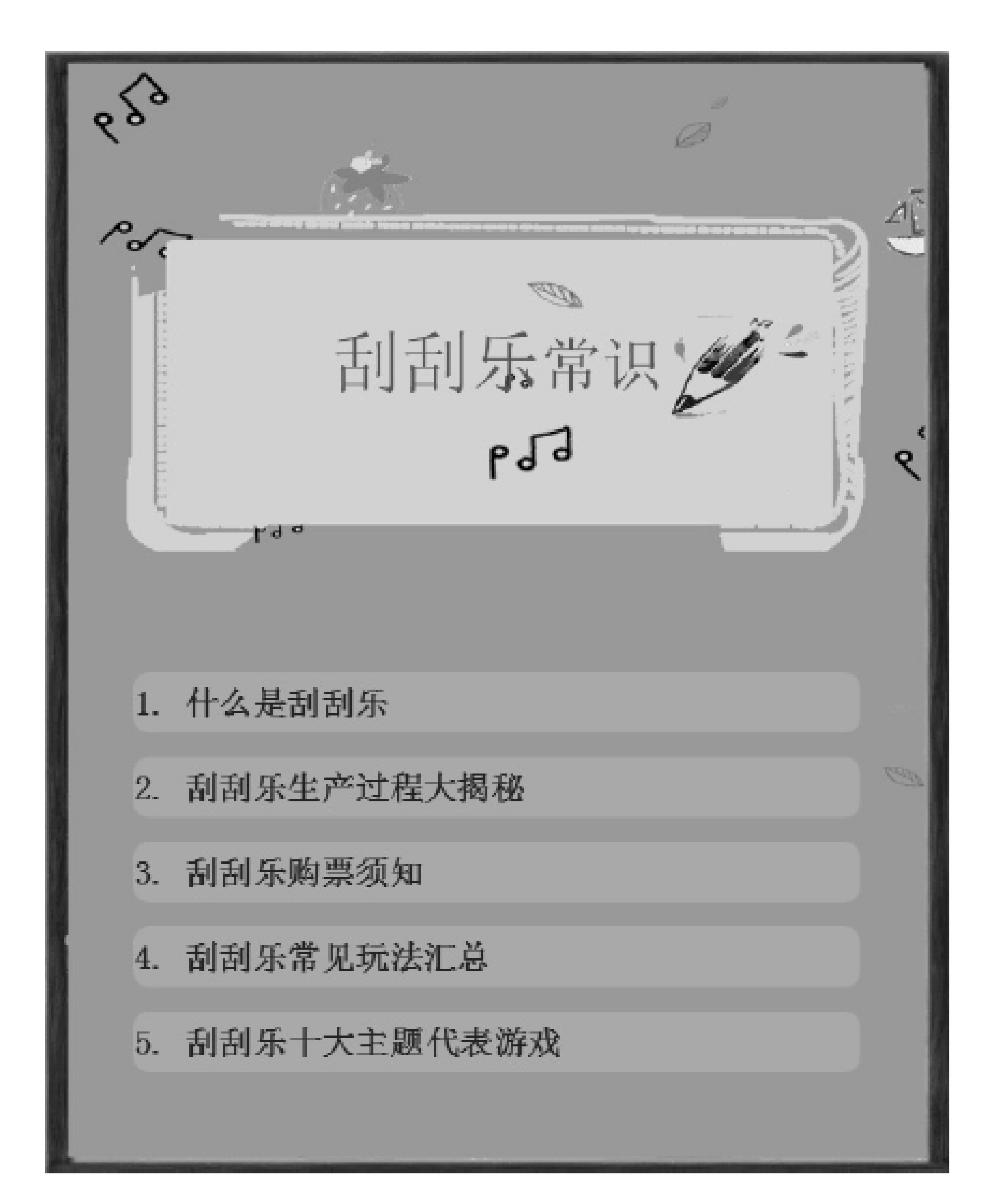


图 7.1　列表中嵌套有序列表

实战技能强化训练

训练一：基本功强化训练

1. 手机详情页面信息　　▷①②③④⑤⑥

网上购物时，在商品详情页面可以查看商品的详细信息。如图 7.2 所示为手机商品的详情页面，请完成该页面。

（提示：颜色、套装、数量等整齐罗列的内容，可通过无序列表实现）

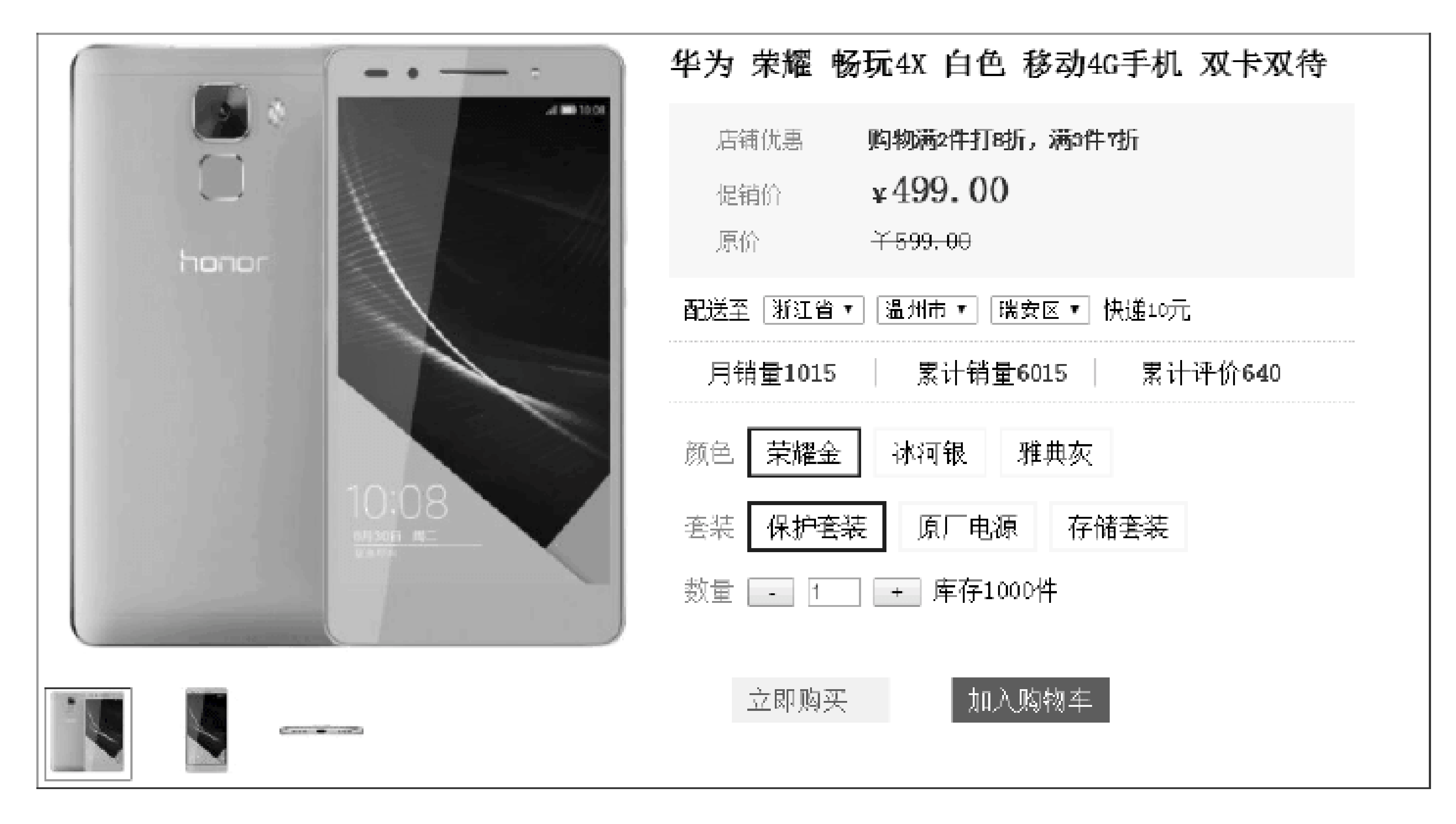
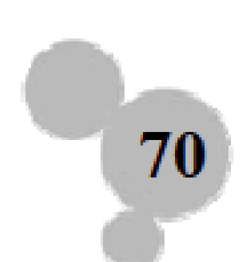


图 7.2　手机详情页面



2. CSS 制作精美书签　▷①②③④⑤⑥

使用 CSS 可以设计出许多精美的书签样式，试着设计如图 7.3 所示的书签效果。（提示：书签上的胶纸效果可通过:after 或者:before 添加，也可以通过<div>实现）

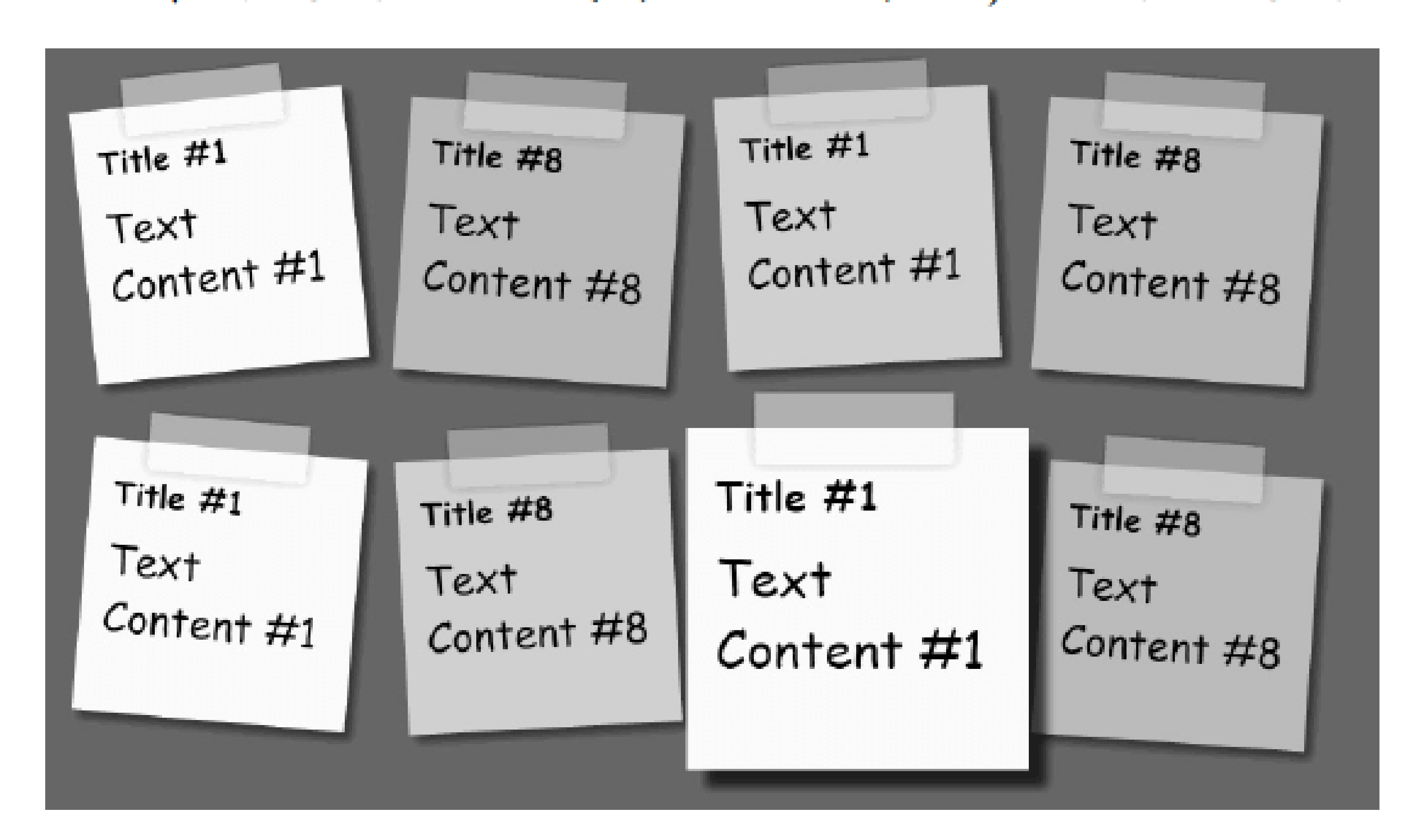


图 7.3　CSS 制作精美书签

3. 餐厅菜单页面　▷①②③④⑤⑥

菜单是一个饭店必不可少的，试着设计一个创意菜单，页面效果如图 7.4 所示。（提示：菜单左上角的圆圈可通过 border-radius 实现）

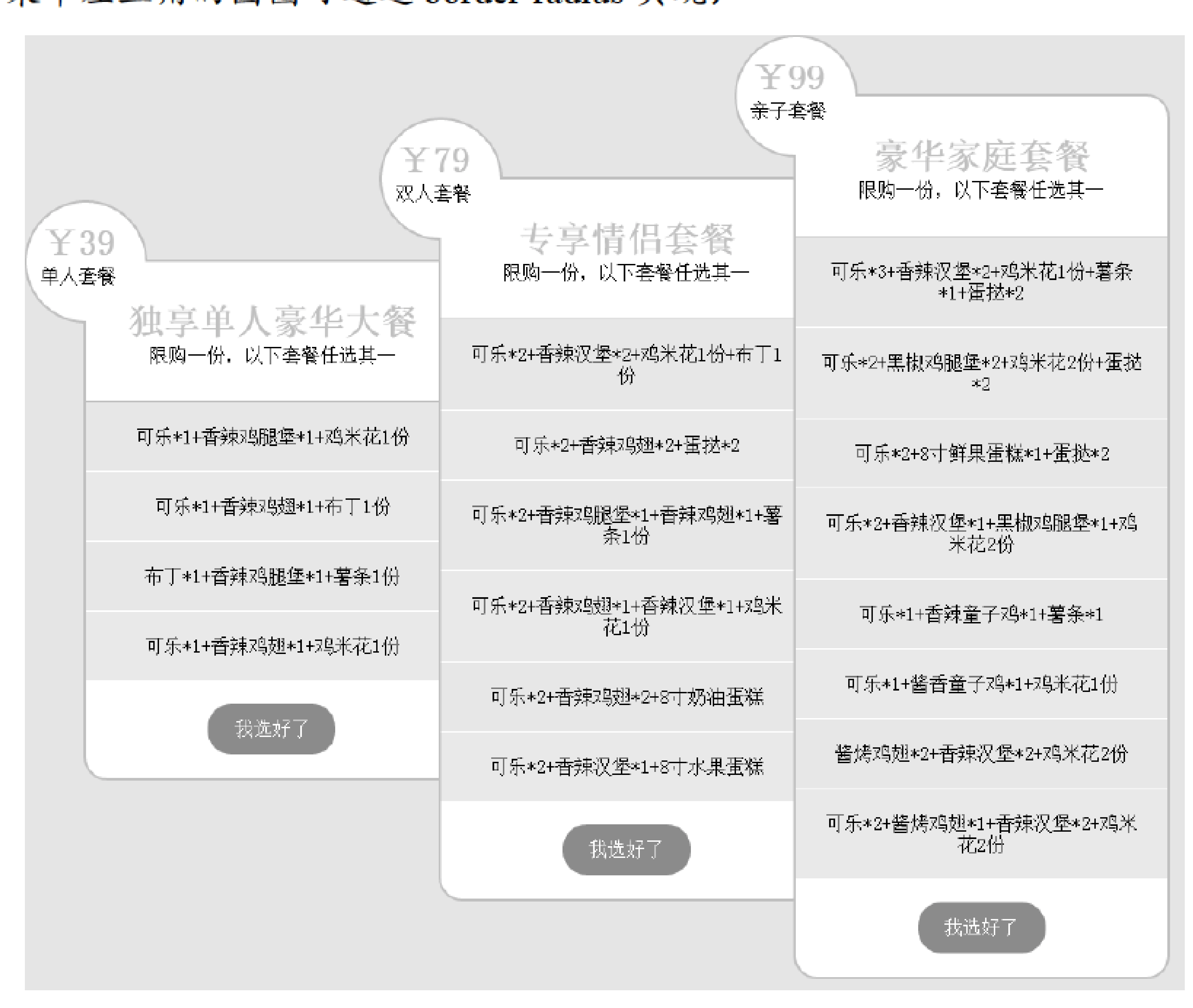


图 7.4　餐厅菜单页面

4．支付宝话费充值页面　▷①②③④⑤⑥

在支付宝上缴纳手机话费，可省去在营业厅排队的等待时间，非常方便。如图 7.5 所示为仿支付宝话费充值页面，请实现该页面。

5．看图猜成语页面　▷①②③④⑤⑥

看图猜成语是一款老少皆宜的益智游戏。编写程序，实现一个看图猜成语游戏页面，效果如图 7.6 所示。



图 7.5　支付宝充值页面



图 7.6　看图猜成语页面

6．修改列表项标志　▷①②③④⑤⑥

可以自定义列表标志，使用 CSS3 中的列表属性还可以自定义列表项图标。如图 7.7 所示页面中，左侧导航栏图标为蓝色小圆圈；当鼠标悬停时，变为绿色小圆圈。请实现该页面。



图 7.7　修改列表项标志

7. QQ 联系人列表　▷①②③④⑤⑥

相信每个人手机都安装有 QQ。如图 7.8 所示为 QQ 联系人列表，请尝试制作这个页面。



图 7.8　QQ 联系人列表

8. 商品热销排行榜　▷①②③④⑤⑥

电商网站中经常会有各类热销商品排行榜，如图 7.9 所示就是一个简单的隐形眼镜热销排行榜页面，试着实现该页面。

9. 票据报销网络填单　▷①②③④⑤⑥

如图 7.10 所示为移动端票据报销填写页面，请实现该页面。

10. 仿手机联系人页面　▷①②③④⑤⑥

实现如图 7.11 所示的仿手机通讯录页面。该页面中，当鼠标指针悬停在联系人分类上时，其子菜单会自动展开。



图 7.9　热销排行榜



图 7.10　票据报销页面



图 7.11　手机联系人列表

训练二：实战能力强化训练

11．分类导航页面　▷①②③④⑤⑥

随着 IT 行业的发展，一些开发学习类网站越来越受欢迎。程序员们可以在这些学习网上寻找自己需要的视频、课程、资源等。制作一个开发学习网站的分类导航页面，当鼠标指针悬停在左侧菜单上时，右侧会展开对应的内容。如图 7.12 所示为鼠标悬停在“网络课程”菜单项时，右侧展开的相应内容页面。



图 7.12　分类导航页面

12．列表实现汽车网站导航菜单　▷①②③④⑤⑥

如图 7.13 所示为列表实现的汽车网站导航菜单。当鼠标指针悬停时，导航菜单下方会显示滑块，并展开二级导航菜单，请实现该页面。

汽车饰品　电子导航　大牌配件　改装装潢　美容保养

行车导航
影音娱乐
车载电器
智能车机
记录仪

图 7.13　列表实现汽车网站导航菜单

13. 红包领取记录　▷①②③④⑤⑥

相信大家都在微信中发过红包，也领过红包。编写程序，实现一个红包领取记录页面，效果如图 7.14 所示。

14. 驾考宝典答题页面　▷①②③④⑤⑥

考驾照分为两部分：理论知识考试和实际操作考试。理论部分，人们可以在“驾考宝典”等 App 上快速做题，模拟考试场景。如图 7.15 所示为“驾考宝典”上科目一的答题页面，请完成该页面。



图 7.14　红包领取记录页面



图 7.15　驾考宝典答题页面

15．教师节节日贺卡 ▷①②③④⑤⑥

教师节要到了，试着制作一份网页版贺卡送给老师，页面效果如图 7.16 所示。

（提示：为图片设置边框可通过 CSS 中的 border-image 实现）



图 7.16　教师节贺卡

16．限时抢购页面 ▷①②③④⑤⑥

编写一个程序，实现购物网站的限时抢购页面，如图 7.17 所示。



图 7.17　限时抢购页面

17．手机商城“热卖推荐”页面 ▷①②③④⑤⑥

实现手机商城中的“热卖推荐”页面。当鼠标指针悬停在某个商品列表上时，商品图片被放大，并且显示出已选参数等信息。如图 7.18 所示为选中第二个商品后的详情效果。

（提示：可以结合伪类选择器与 display 属性设置商品详情的显示与隐藏）

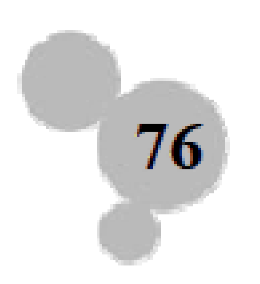




图 7.18　手机商城中热卖推荐

18．开发类网站的二级导航菜单　▷①②③④⑤⑥

编写一段程序，制作软件开发类网站的二级导航菜单，具体效果如图 7.19 所示。



图 7.19　自学网站的二级导航菜单

19．轮播介绍手机的儿童模式　▷①②③④⑤⑥

部分手机可以设置儿童模式。开启儿童模式后，可以防沉迷，限制游戏、应用等开启的时间，保护儿童稚嫩的视力。编写程序，通过轮播图的自动切换介绍儿童模式，页面效果如图 7.20 所示。



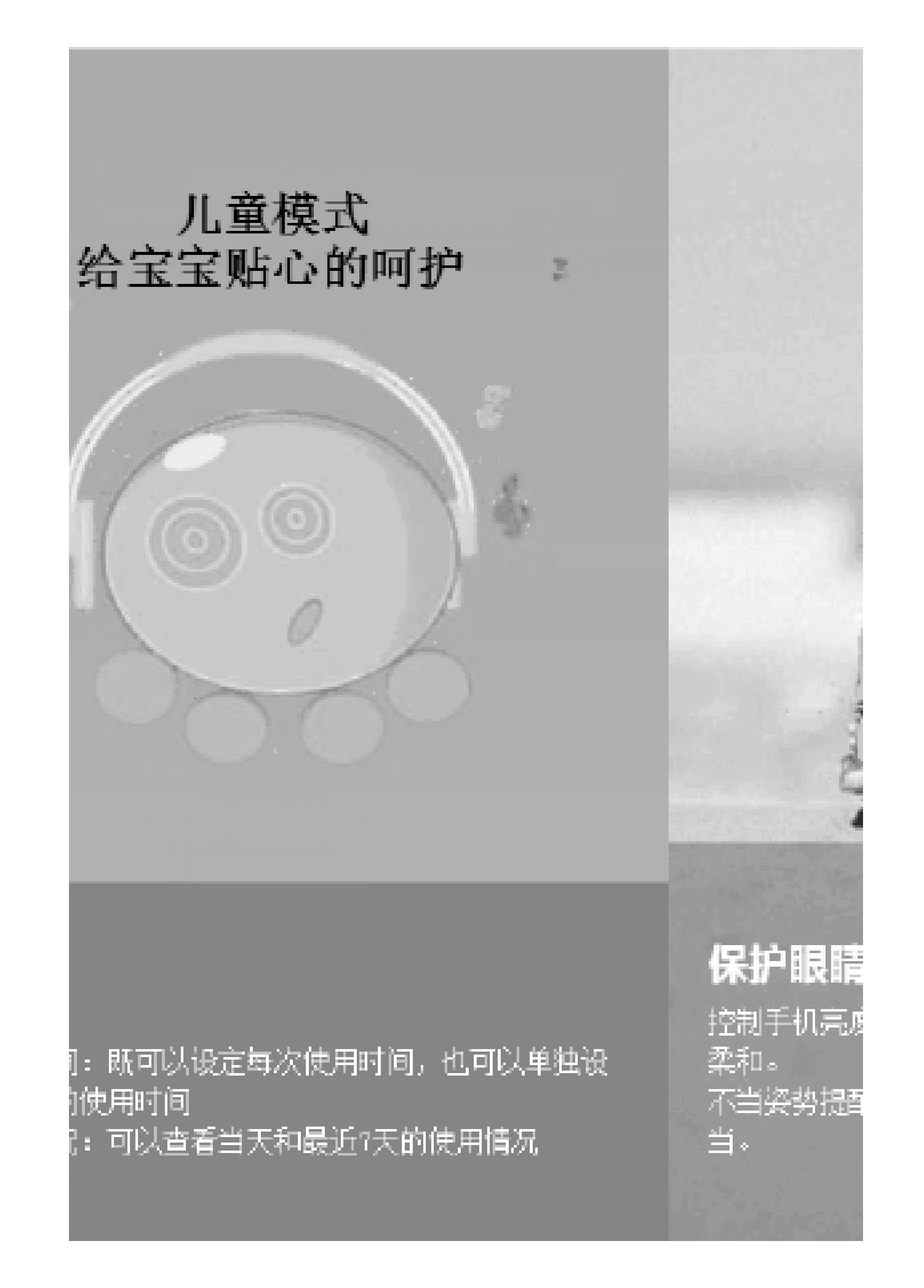


图 7.20　手机儿童模式介绍

20．时间轴显示各月份最适合的景点　▷①②③④⑤⑥

编写程序，通过一个时间轴显示各月份最适合旅游的景点，页面效果如图 7.21 所示。

（提示：时间轴首、末端的圆和箭头既可以通过 div 实现，也可以通过:after 和:before 实现）

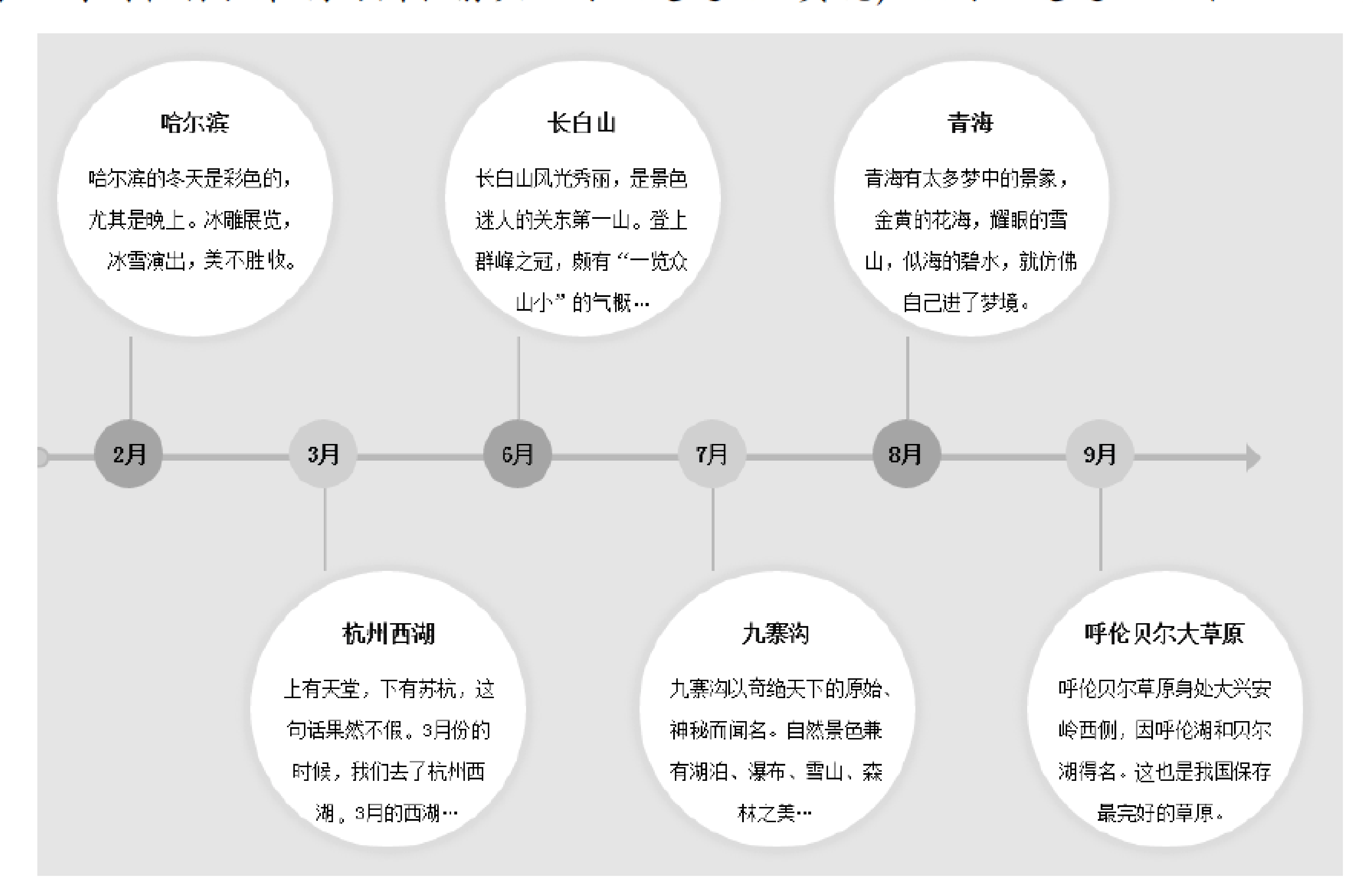


图 7.21　时间轴显示各月份最适合的景点

第 8 章　表　　单



本章训练任务对应核心技术分册第 8 章“表单”部分。

重点练习内容：

1. 表单标签的使用。
2. input标签中type属性各属性值的应用。
3. 合理布局表单页面。
4. 掌握input标签的其他常用属性。

应用技能拓展学习

1. HTML5 新增的 input 标签的属性

input 标签中，type 属性可以根据不同的属性值，定义不同的输入字段类型。type 属性值不同，其功能也不同，如 type="text"和 type="password"等。除此之外，input 标签还新增了许多属性，如表 8.1 所示。（由于 type 属性大家已熟知，故表中没有列举）

表 8.1　input 标签新增的属性及其含义

属　性	属　性　值	含　义	说　明
accept	audio/* video/* image/*等	规定通过文件上传来提交的文件类型	只针对 type="file"
alt	text	定义图像输入的替代文本	只针对 type="image"
autocomplete	on/off	规定是否开启自动完成	
autofocus	autofocus	页面加载时，<input>标签自动获取焦点	
checked	checked	页面加载时，<input>标签被预先选中	只针对 type="checkbox"或 type="radio"
disabled	disabled	禁用该<input>标签	
form	form_id	规定<input>标签所属的一个或多个表单	
formaction	url	表单提交时处理输入控件的文件 URL	只针对 type="submit"和 type="image"
formenctype	Application text/plain 等	表单提交到服务器时如何编码	只针对 type="submit"和 type="image"
formmethod	get post	定义发送表单数据到 actionURL 的 HTTP 方法	只针对 type="submit"和 type="image"
formnovalidate	formnovalidate	覆盖<form>标签的 novalidate 属性	

续表

属　性	属　性　值	含　义	说　明
formtarget	_blank _self _top _parent	表单提交后，显示接收到的相应名称或关键字	只针对 type="submit"和 type="image"
height	pixels	规定<input>元素的高度	只针对 type="image"
list	datalist_id	应用<datalist>标签，其中包含<input>标签预定义的选项	
max	number	规定<input>标签的最大值	
maxlength	number	规定<input>的最大字符数	
min	number	规定<input>标签的最小值	
nultiple	nultiple	允许用户输入多个值	
name	text	定义<input>元素的名称	
pattern	regexp	规定用于验证<input>值的正则表达式	
placeholder	text	规定<input>标签预期值的提示信息	
readonly	readonly	规定输入字段为只读	
required	required	规定提交表单之前必须输入字段	
size	number	规定<input>标签的可见宽度，单位为字符	
src	URL	规定提交按钮的图像的 URL	只针对 type="image"
step	number	规定<input>标签的合法数字间隔	
value	text	规定<input>中 value 的值	
width	pixles	规定<input>元素的宽度	只针对 type="image"

例如，通过<datalist>标签和 list 属性为文本框添加快捷输入选项。代码如下：

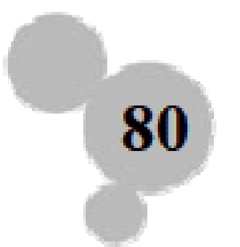
```
<form>
    <label>请选择编程语言</label><input type="text" list="list1">
    <datalist id="list1">
        <option value="HTML">
        <option value="Java">
        <option value="PHP">
        <option value="Android">
        <option value="python">
        <option value="C#">
    </datalist>
</form>
```

其实现效果如图 8.1 所示。



图 8.1　list 属性的使用

（提示：表 8.1 所列举的属性中，有的可以直接使用，有的则需要结合其他标签使用。例如，list 属性需要结合<datalist>标签使用）



实战技能强化训练

训练一：基本功强化训练

1．QQ 注册页面　▷①②③④⑤⑥

QQ 是我们常用的聊天工具之一，甚至很多人注册了两个以上 QQ 账号。如图 8.2 所示为仿 QQ 注册页面，请实现该页面。



图 8.2　QQ 注册页面

2．电子发票开具页面　▷①②③④⑤⑥

开发票大家都经历过，那么大家有没有开具过电子发票呢？如图 8.3 所示为开具电子发票的页面，请实现该页面。

3．中奖信息填写页面　▷①②③④⑤⑥

一些网站为了促销，会不定时地进行抽奖或派发优惠券。中奖的小伙伴需要先绑定账号，然后才能兑奖。编写一个程序，实现中奖信息填写页面。



图 8.3　电子发票开具页面



图 8.4　中奖信息填写页面

4．简约登录页面　▷①②③④⑤⑥

网站的功能不同，其页面风格也不同。如图 8.5 所示就是一个简约风格的登录页面，请实现该页面。

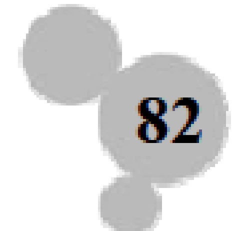


图 8.5　简约登录页面

5．“爱家”在线租房申请页面　▷①②③④⑤⑥

多数人都有过租房的经历，制作如图 8.6 所示的“爱家”在线租房申请页面。

（提示：通过 CSS 中的-webkit-linear-gradient 属性实现页面背景的线性渐变）



6．商品评价时限制输入的长度　▷①②③④⑤⑥

很多网站的表单在用户填写时都会限制输入内容的长度，如限制用户名和密码的长度等。实现如图 8.7 所示网购后的商品评价页面，限制输入内容的长度在 40 个字以内。

（提示：通过 maxlength 属性设置文本域内容的长度）



图 8.6　“爱家”在线租房申请页面



图 8.7　限制输入的长度

7．玩转漂流瓶　▷①②③④⑤⑥

漂流瓶是 QQ 和微信中早期的功能，模拟实现如图 8.8 所示的漂流瓶页面效果。

8．在线留言页面　▷①②③④⑤⑥

编写程序，实现移动端在线留言页面，具体效果如图 8.9 所示。

（提示：“在线留言”两侧的菱形块可通过:after 和:before 实现，也可通过添加 DIV+ CSS 实现）



图 8.8　玩转漂流瓶

9．包含第三方登录接口的会员登录页面　▷①②③④⑤⑥

登录页面很常见，这里设计一个包含第三方登录接口的登录页面，效果如图 8.10 所示。



图 8.9　在线留言页面



图 8.10　包含第三方信息的登录页面

10．文本框不为空验证　▷①②③④⑤⑥

注册或登录账号时，某些关键项没有填写完整，就无法完成登录或注册，并且会弹出不能为空的提示。使用 HTML 中的属性可以完成表单的非空验证。如图 8.11 所示，当用户昵称、密码以及确认密码为空时，提交该页面，就会弹出非空警告，请实现该页面。

（提示：非空验证可以通过 HTML 中的 required 属性实现）

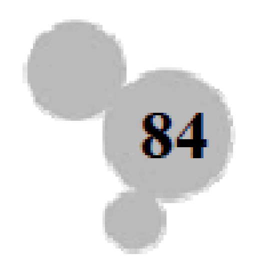




图 8.11　文本框不为空验证

训练二：实战能力强化训练

11．手机端购票页面　▷①②③④⑤⑥

编写程序，实现手机端用户购票时的订单信息页面，效果如图 8.12 所示。

12．个人档案页面　▷①②③④⑤⑥

几乎所有网站在用户注册时，都会搜集并保存用户上传的个人信息。如图 8.13 所示为用户个人档案填写页面，请实现该页面。

图 8.12　手机端购票页面

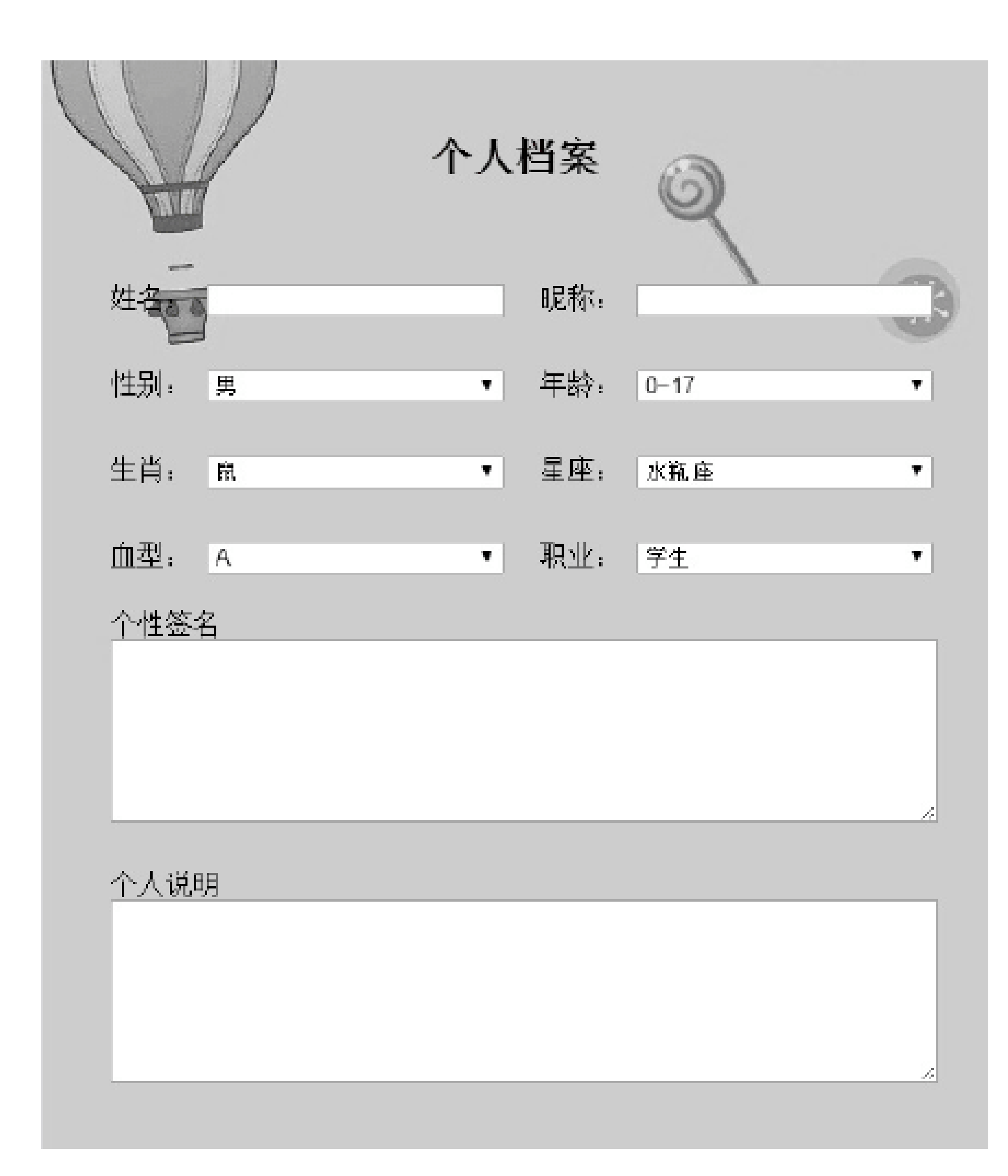


图 8.13　个人档案填写页面

13．百货超市满意度调查表 ▷①②③④⑤⑥

制作一个百货超市的满意度调查表页面，其实现效果如图 8.14 所示。

百货超市满意度调查表

姓名：　　性别：　○男　○女

1、是否经常光顾我店

○是的　○不，去过一两次　○没去过

2、您觉得我店的商品质量怎么样

○很不好　○一般　○还不错　○不了解

3、对于我店，您最满意的是？

□服务态度　□商品质量　□售后服务　□店内环境

4、对于我店，您最不满意的是？

□服务态度　□商品质量　□售后服务　□店内环境

5、您日常消费在哪些方面？

□服装　□吃　□健身　□生活用品

6、消费时，你最注重店里的

□商品质量　□价格便宜　□交通便利　□服务态度

图 8.14　百货超市满意度调查表

14．QQ 空间留言板 ▷①②③④⑤⑥

制作一个 QQ 空间留言板，并且美化该页面，具体实现效果如图 8.15 所示。

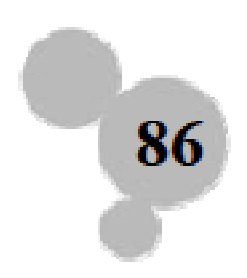




图 8.15　QQ 空间留言板

15．大连一日游预定报名页面　▷①②③④⑤⑥

以前逢年过节，家人通常会团聚在一起，聚会娱乐。而现在，许多人们会选择全家一起外出旅游。如图 8.16 所示为大连一日游的预订报名页面，请完成该页面。



图 8.16　旅游报名页面

16．bug 意见反馈页面

▷①②③④⑤⑥

写程序出现 bug 是常有的事情。如图 8.17 所示为某进销存管理系统的 bug 意见反馈页面，请实现该页面。



图 8.17　bug 意见反馈页面

第 9 章　多　媒　体



本章训练任务对应核心技术分册第 9 章“多媒体”部分。

重点练习内容：

1. Video与audio的简单使用。
2. Video对象与audio对象中常用的属性和方法。
3. 自定义视频和音频的工具栏。
4. 获取和设置视频及音频的状态、时长等常用数据。

应用技能拓展学习

1. Audio 标签与 Video 标签的使用

Audio 标签（即<audio>标签）和 Video 标签（即<video>标签）是 HTML5 中新增的标签。其中，<audio>标签用于添加音频，<video>标签用于添加视频。有了这两个标签，播放视频时就不再需要插件了。

<audio>标签的使用语法如下：

```
<audio src="media/6.mp3" controls autoplay loop></audio>
```

☑ src：表示待添加的音频的路径。

☑ controls：表示显示播放器的控制栏。此为简写形式，具体写法是：controls="controls"。

☑ autoplay：表示音乐加载结束后自动播放。此为简写形式，具体写法是：autoplay ="autoplay"。

☑ loop：表示音乐循环播放。此为简写形式，具体写法是：loop="loop"。

<video>标签的使用语法如下：

```
<video src="media/2.mp4" poster="media/96.png" autoplay controls loop></video>
```

这里，src、controls、autoplay 等属性的含义与<audio>标签相同，poster 属性用于定义视频加载时（或播放前）显示的图像。

2. Audio 对象和 Video 对象的属性

自定义视频或音频工具栏后，要实现工具栏中按钮的功能，则需要通过 Audio 对象和 Video 对象的属性来实现。

Audio 对象和 Video 对象的属性大多类似，一些常用的属性及其含义如表 9.1 所示。

表 9.1　Audio 对象和 Video 对象的属性及含义

属　　性	说　　明	备　　注
audioTacks	返回可用于音频轨道的 AudioTrackList 对象	
autoplay	设置或返回是否在加载完成后随机播放音频（视频）	
buffered	返回表示音频（视频）已缓冲部分的 TimeRanges 对象	
controller	返回表示音频（视频）当前媒体控制器 MediaController 对象	
controls	设置或返回音频（视频）是否显示控件（比如播放/暂停等）	
crossOrigin	设置或返回音频（视频）的 CORS 设置	
currentTime	设置或返回音频（视频）当前的播放位置（单位 s）	
currentSrc	返回当前音频（视频）的 URL	
defauleMuted	设置或返回音频（视频）是否默认为静音	
defaultPlaybackRate	设置或返回音频（视频）的默认播放速度	
duration	返回当前音频（视频）的长度（单位 s）	
ended	返回音频（视频）播放是否已结束	
error	返回音频（视频）错误状态的 MediaError 对象	
height	设置或返回 height 属性的值	仅针对 video 对象
loop	设置或返回音频（视频）是否应在结束时重新播放	
mediaGroup	设置或返回音频（视频）是否静音（用于连接多个音频元素）	
muted	设置或返回音频（视频）是否静音	
networkState	返回音频（视频）的当前网络状态	
paused	设置或返回音频（视频）是否暂停	
playBackRate	设置或返回音频（视频）播放的速度	
played	表示音频（视频）已播放部分的 TimeRanges 对象	
poster	设置或返回 poster 属性的值	仅针对 video 对象
preload	设置或返回音频（视频）是否在页面加载完后进行预加载	
readyState	表示音频（视频）当前的就绪状态	
seekable	返回音频（视频）可寻址部分的 TimeRanges 对象	
seeking	返回用户当前是否正在音频（视频）中查找	
src	设置或返回音频（视频）src 属性的值	
startDate	返回当前时间偏移的 Date 对象	仅针对 video 对象
textTracks	返回可用文本轨道的 TextTrackList 对象	
volume	设置或返回视频（音频）的音量	
width	设置或返回视频 width 属性的值	仅针对 video 对象

3. Audio 对象和 Video 对象方法

除了属性以外，为视频、音频添加某些功能时还需要用到一些方法。表 9.2 列举了 Audio 和 Video 对象常用的方法，综合使用这些方法和属性，可以实现一些意想不到的功能。

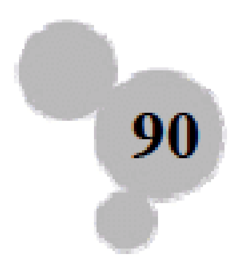


表 9.2　Audio 对象和 Video 对象方法

方　法	说　明
addPlayTracke()	向视频（音频）添加新的文本轨道
canPlayType()	检查浏览器是否能够播放指定的视频（音频）类型
load()	重新加载视频元素
play()	开始播放视频
pause()	暂停当前播放的视频

例如，在网页中添加视频，定义“暂停”和“播放”两个按钮，且视频暂停时显示视频的总时长及播放位置。代码如下：

```
<div style="text-align: center">
    <video src="media/3.mp4" id="video1"></video><br>
    <button type="button" onclick="play1()">播放</button>
    <button type="button" onclick="pause1()">暂停</button>
</div>
<script type="text/javascript">
    var video1 = document.getElementById("video1");
    function play1() {
        video1.play();
    }
    function pause1() {
        video1.pause();
        alert("已暂停,该视频总时长为" + video1.duration + "s，已播放" + video1.currentTime + "s")
    }
</script>
```

其运行结果如图 9.1 所示。



图 9.1　视频暂停时获取相关信息

实战技能强化训练

训练一：基本功强化训练

1．网页中添加视频　　▷①②③④⑤⑥

爱奇艺、优酷等视频网站深受人们喜爱，因为里面有许多精彩大片。编写程序，使用<video>标签在网页中添加一段视频，运行效果如图 9.2 所示。

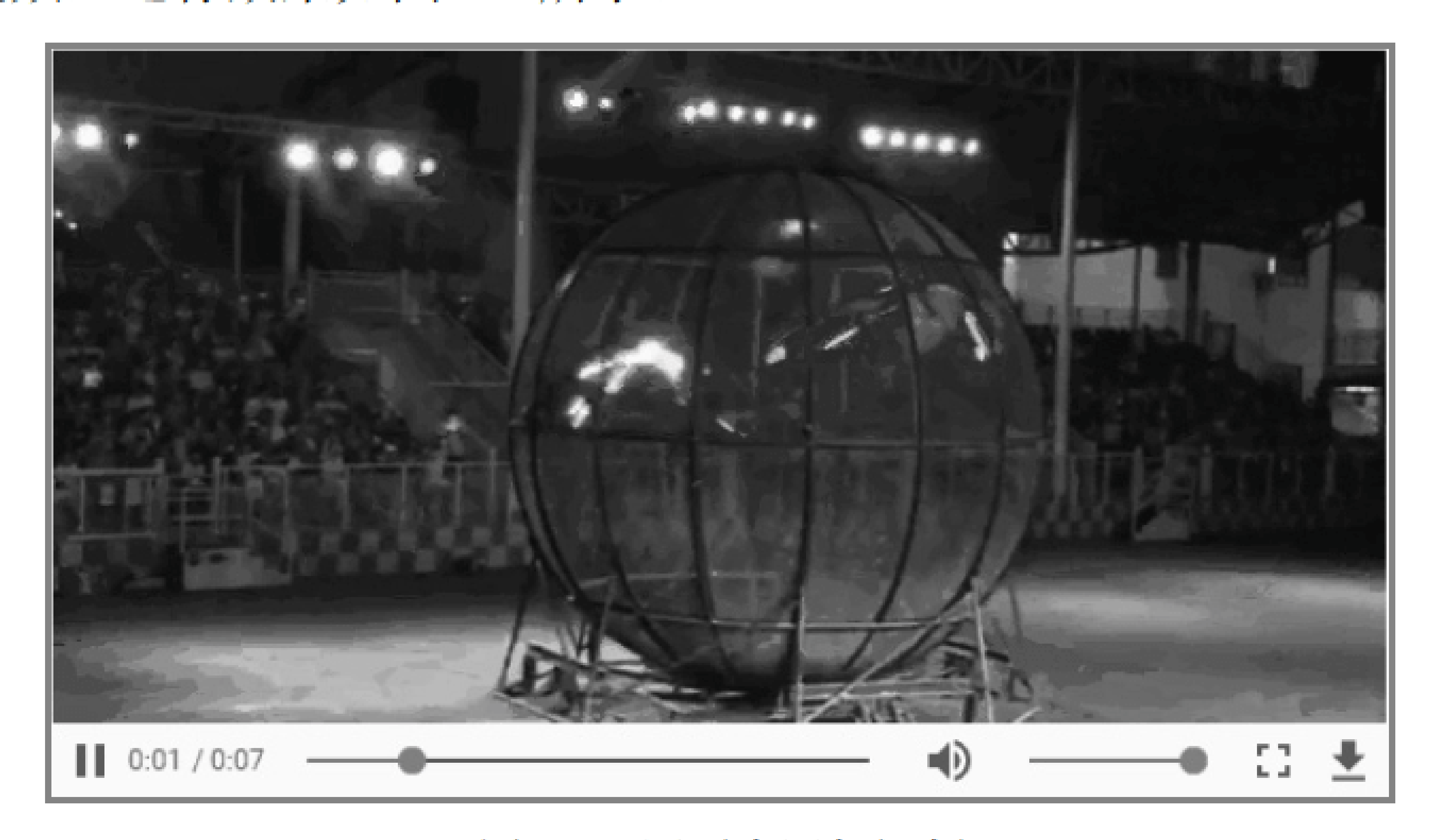


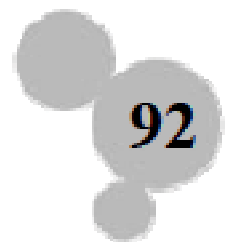
图 9.2　网页中添加视频

2．实现直播页面　　▷①②③④⑤⑥

直播是近几年来兴起的一个热门行业。编程实现一个直播页面，运行效果如图 9.3 所示。



图 9.3　直播页面



3．设置视频控制栏的显示与隐藏　▷①②③④⑤⑥

在网页中添加视频，设置视频播放前的显示图像，并动态显示与隐藏视频控制栏，具体效果如图 9.4 与图 9.5 所示。

（提示：设置视频播放前显示的图像可以通过 poster 属性，其属性值为所设置图像的路径）



图 9.4　隐藏控制栏



图 9.5　显示控制栏

4．添加背景音乐　▷①②③④⑤⑥

编写程序，在网页中添加背景音乐，设置页面背景颜色随时间发生变化，并且歌词内容和歌曲信息互相切换显示，其实现效果如图 9.6 和图 9.7 所示。

（提示：判断音频的播放状态可以通过 paused 属性，该属性用于判断音频是否被暂停）

图 9.6　显示歌词内容

图 9.7　切换歌曲时背景发生变化

5．动态设置视频大小与播放状态　　▷①②③④⑤⑥

我们在观看视频时，可以通过控制栏设置视频的状态。其实，这些控制栏并非是固定不变的，用户也可以自定义视频的控制栏。如图 9.8 所示，自定义视频的播放与暂停按钮，并且通过按钮实现视频的放大与缩小。

（提示：设置视频的大小可以通过 video 对象中的 width 和 height 属性实现）

6．自定义视频工具栏　　▷①②③④⑤⑥

编写一个程序，自定义视频的工具栏，其运行效果如图 9.9 所示。通过控制栏中的◁ ⏸ ▷按钮可以实现视频的播放与暂停，同时显示视频的播放进度以及视频总时长。

（提示：获取视频总时长可以通过 video 对象的 duration 属性；而获取视频的播放进度可以通过 currentTime 属性）



图 9.8　实现效果

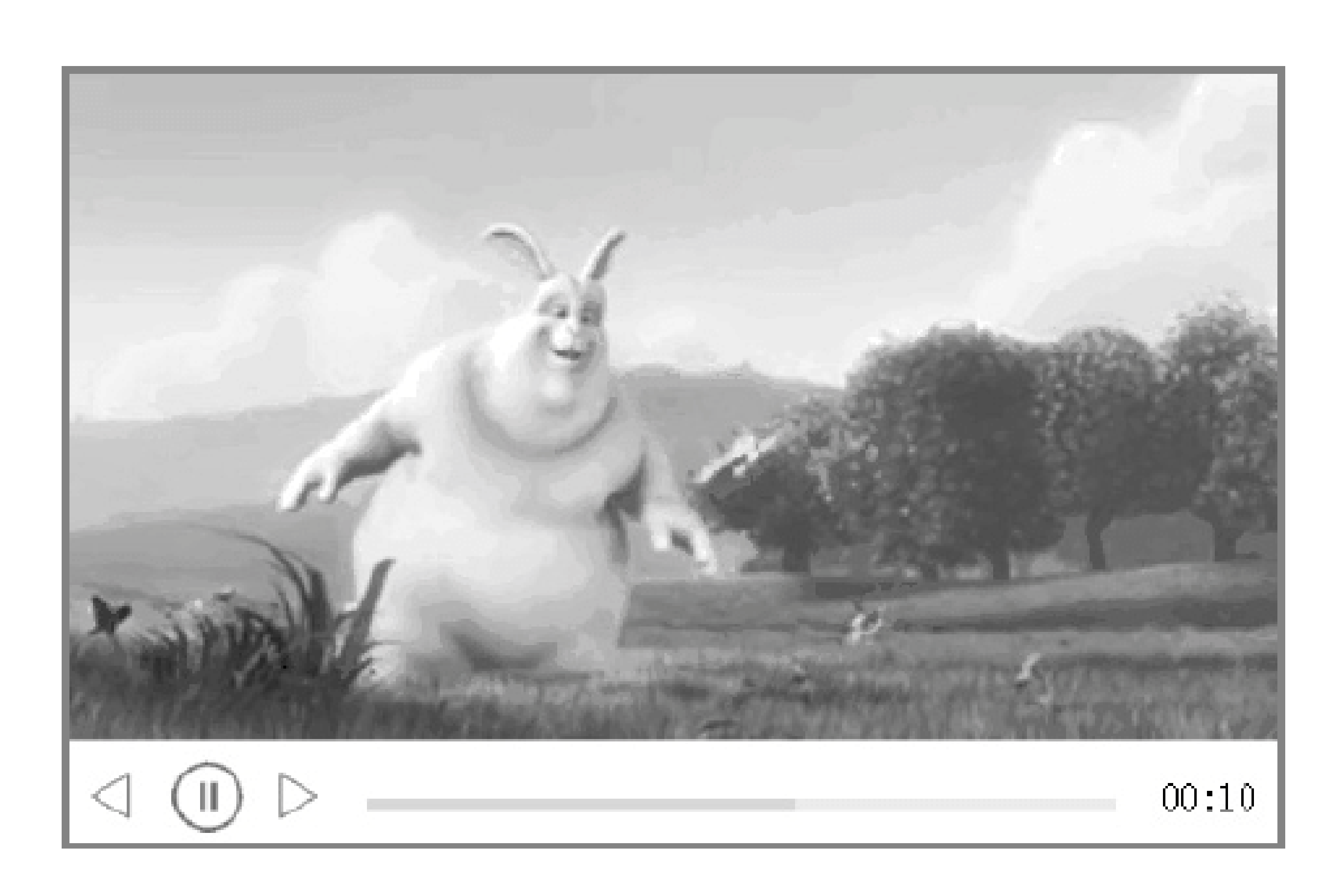


图 9.9　实现效果

7. 音乐循环播放　▷①②③④⑤⑥

实现一款音乐播放器，音乐可暂停播放和显示播放进度，还可以动态设置音乐是否循环播放以及设置音乐播放动画。其运行效果如图 9.10 所示。

（提示：可以通过 audio 对象的 loop 属性获取和设置循环播放音频）

8. HTML5 手机视频播放器　▷①②③④⑤⑥

编写一个程序，模拟在手机中播放视频的效果。设置鼠标悬停在视频上时，上方滑出视频标题，右侧滑出分享和收藏等按钮。视频播放时，显示播放进度，并可以设置视频暂停播放、继续播放、放大、缩小等效果。实现效果如图 9.11 所示。

（提示：鼠标悬停和离开视频时调用函数可以通过 onmouseover 和 onmouseout 实现）



图 9.10　设置音频是否循环播放



图 9.11　仿手机播放器

9. 实现音乐、歌词同步 ▷①②③④⑤⑥

编写程序，为音乐添加歌词，且歌词与播放进度同步显示，运行效果如图 9.12 所示。

（提示：在 HTML 中添加歌词，然后通过 JavaScript 将歌词中的时间与歌词内容分割，最后将时间与播放进度进行比较，实现歌词同步）

10. 为视频添加弹幕 ▷①②③④⑤⑥

看视频或者直播的时候，经常会看到许多弹幕文字。编写程序，实现弹幕发送功能，效果如图 9.13 所示。在文本框中输入内容，然后提交，弹幕内容就会在视频中随机显示并向左移动；移除视频范围后，弹幕就会被删除。

（提示：动态添加字幕，需要通过 getCreatElement()方法新建元素，然后通过 appendChild()方法将新建元素添加到父元素中）



图 9.12 播放音乐时同步显示歌词



图 9.13 为视频添加弹幕

训练二：实战能力强化训练

11. 为视频添加字幕 ▷①②③④⑤⑥

编写程序，为视频添加字幕，具体实现效果如图 9.14 所示。

（提示：为视频添加字幕与为音频添加歌词的原理类似）

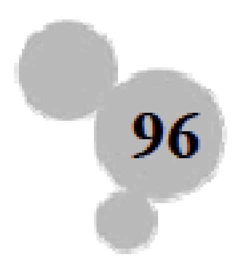




图 9.14　为视频添加字幕

12．实现 PC 端音乐播放器　▷①②③④⑤⑥

实现一款 PC 端音乐播放器，控制栏效果如图 9.15 所示。可以实现前后歌曲切换以及暂停和播放歌曲，可以增减歌曲播放时的音量。单击控制栏中的列表按钮时，控制栏下方会显示歌曲列表，如图 9.16 所示。

（提示：设置音量大小可以通过 audio 对象的 column 属性实现）

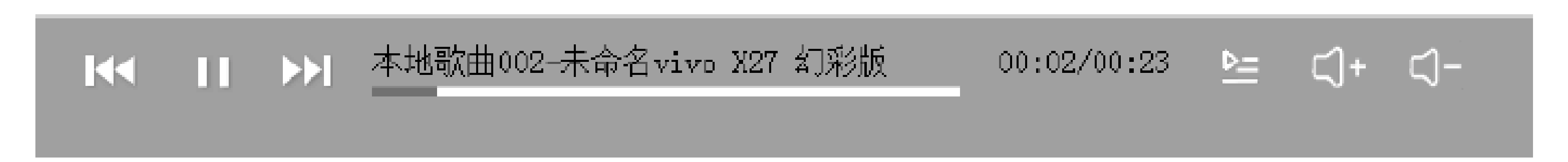


图 9.15　音乐播放器控制栏

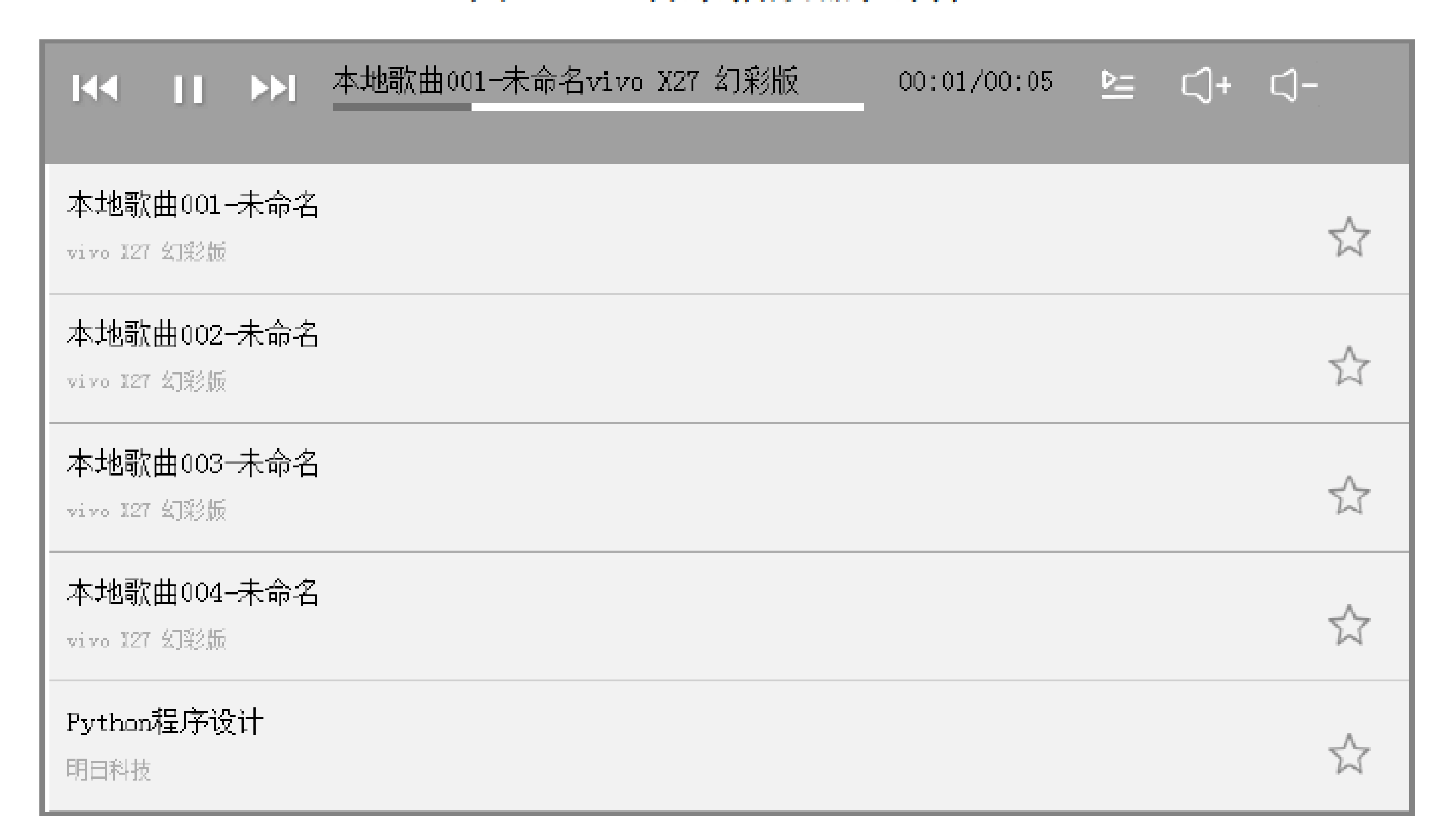


图 9.16　显示歌曲列表

13．模拟游戏音效　▷①②③④⑤⑥

玩游戏的时候，游戏中精彩的音效可以增强玩家的体验感和浸入感。制作一个小游戏，当小球碰

撞到四周时，触发碰撞音效，如图 9.17 所示。

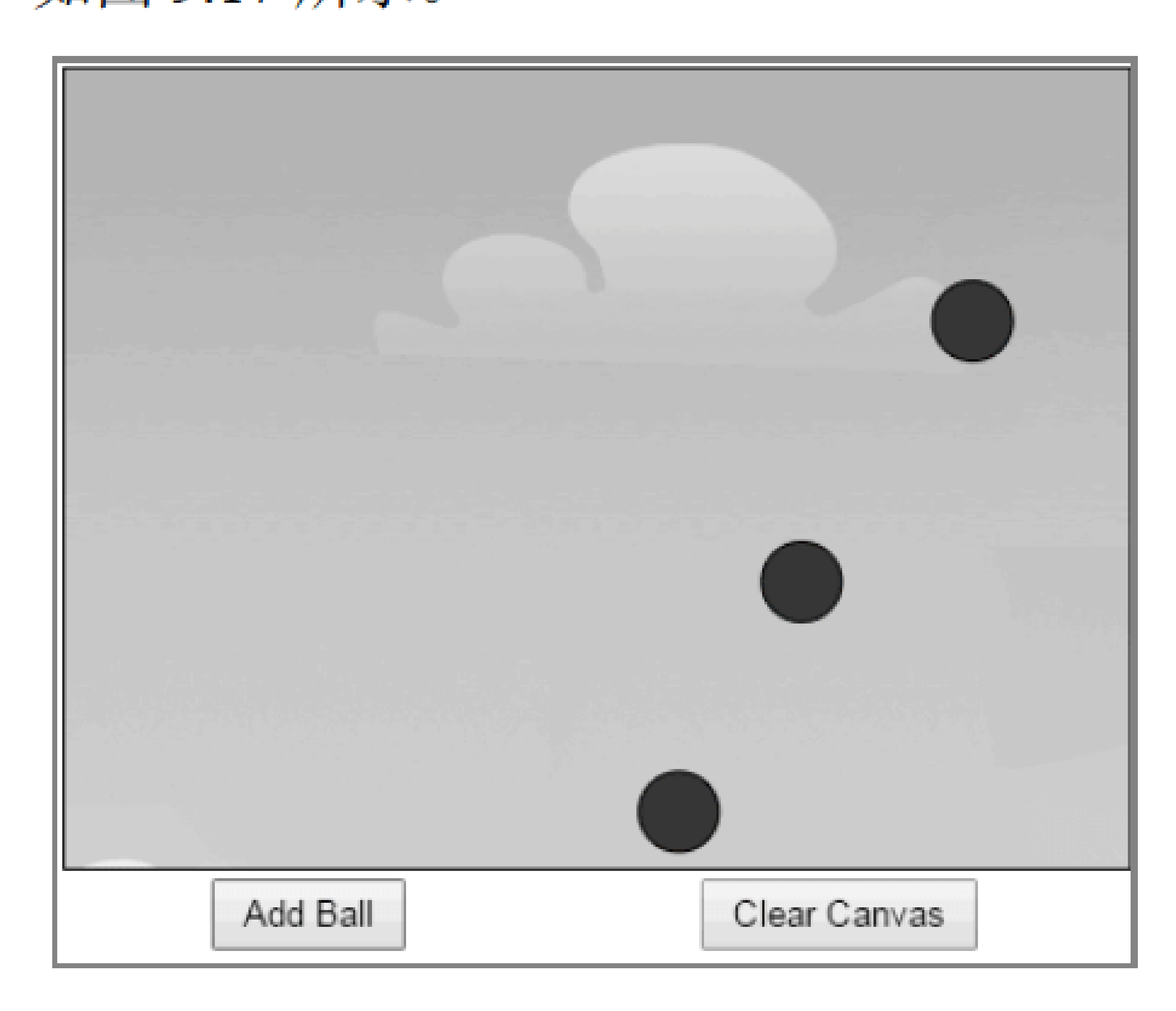


图 9.17　模拟游戏音效

14．仿酷狗音乐播放器　▷①②③④⑤⑥

酷狗是一款广受欢迎的音乐 APP，下面模仿酷狗音乐播放器，实现音乐播放功能，效果如图 9.18 所示。单击歌曲列表中的播放/暂停按钮，可播放或暂停音乐；单击下方工具栏中的左右切换按钮，可以切换音乐。



图 9.18　仿酷狗音乐播放器

第 10 章　HTML5 新特性



本章训练任务对应核心技术分册第 10 章“HTML5 新特性”部分。

重点练习内容：

1. 了解语义化元素的含义。
2. 知道HTML5中新增的元素及其应用。
3. 掌握HTML5中新增的表单属性。
4. 了解HTML5中废弃的元素。

应用技能拓展学习

1. HTML5 中新增的结构元素与语义化元素

每个标签都有固定的含义，如标题标签<hx>、段落标签<p>等。开发网站时，要实现不同的功能，需要使用不同的标签，这就叫作语义化。

语义化的主要好处是对搜索引擎友好。HTML4 网页中，较多地使用了<div>标签，不利于被搜索引擎抓取，所以 HTML5 中新增了一些语义化元素和结构元素。

（1）新增的结构元素

☑ section 元素：定义文档或应用程序的一个区段，如章节、页脚或文档其他部分，可与 h1、h2 等元素结合使用。其语法如下：

```
<section>…</section>
```

☑ article 元素：定义文档中一块独立的内容，如博客或报纸中的一篇文章。其语法如下：

```
<article>...</article>
```

☑ header 元素：定义页面的第一个内容区块或标题。其语法如下：

```
<header>…</header>
```

☑ nav 元素：定义页面的导航部分。其语法如下：

```
<nav>…</nav>
```

☑ footer 元素：定义整个页面或页面中某个内容区块的脚注。一般来说，包含创作者的姓名、文档的创作日期以及创建者联系信息。其语法如下：

```
<footer>…</footer>
```

（2）新增的语义元素

☑ aside 元素：定义 article 元素内容之外但与其内容相关的内容。其语法如下：

```
<aside>…</aside>
```

☑ figure 元素：定义一段独立的流内容，一般表示为文档主体流内容中的一个独立单元。可使用 <figcaption> 元素为 figure 元素组添加标题。其语法如下：

```
<figure>
    <figcaption> 标题 </figcaption>
    <p>文本</p>
</figure>
```

☑ dialog 元素：定义对话（如交谈）。其代码示例如下：

```
<dialog open>
    <dt>小 A:</dt>
    <dd>你知道 ABCDEFG 的含义吗</dd>
    <dt>小 B:</dt>
    <dd>A boy can do everything for girl.</dd>
    <dt>小 B:</dt>
    <dd>可你知道 HIJK 的含义吗</dd>
    <dt>小 A:</dt>
    <dd>不知道</dd>
    <dt>小 B:</dt>
    <dd>He is just kidding</dd>
</dialog>
```

上述代码的运行结果如图 10.1 所示。

小A:你知道ABCDEFG的含义吗
小B:A boy can do everything for girl.
小B:可你知道HIJK的含义吗
小A:不知道
小B:He is just kidding

图 10.1　dialog 会话框

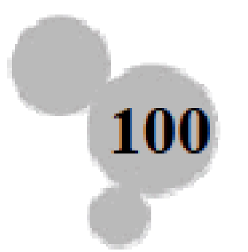
☑ mark 元素：定义需要突出显示或高亮显示的文字。最典型的应用就是在搜索结果中向用户高亮显示搜索关键词。其语法如下：

```
<mark>…</mark>
```

☑ time 元素：定义日期或时间，也可以同时定义两者。其语法如下：

```
<time>…</time>
```

☑ progress 元素：定义运行中的进程，可用来显示 JavaScript 中耗费时间的函数进程。其语法如下：



```
<progress>…</progress>
```

2. HTML5 中新增的 input 元素类型

HTML5 中，新增了很多 input 元素类型。

☑ email：包含 e-mail 地址的输入域。
☑ url：包含 URL 地址的输入域。
☑ number：包含数值的输入域。
☑ range：包含一定范围数字值的输入域。
☑ Date Pickers（数据检出器）。
☑ Search：搜索域（如站点搜索或 Google 搜索等），显示为常规的文本域。

HTML5 拥有多个可供选取日期和时间的新输入类型。

☑ date：选取日、月和年。
☑ month：选取月和年。
☑ week：选取周和年。
☑ time：选取时间（小时和分钟）。
☑ datetime：选取时间、日、月、年（UTC 时间）。
☑ datetime-local：选取时间、日、月、年（本地时间）。

实战技能强化训练

训练一：基本功强化训练

1. 会说话的汤姆猫　▷①②③④⑤⑥

“会说话的汤姆猫”是一款休闲游戏，深受孩子们的喜爱，甚至市场上还有简易版的汤姆猫手机玩具。模拟实现该游戏，当鼠标单击汤姆猫身体不同部位时，汤姆猫可做出不同反应，运行效果如图 10.2 所示。

（提示：通过<map>标签与<area>标签定义图片中的不同部位）

2. 文字边缘镂空动画　▷①②③④⑤⑥

<svg>是 HTML5 中一个新的标签，使用它可实现一些简单的图形和动画。尝试实现如图 10.3 所示的镂空文字边缘动画。

（提示：通过 stroke-dasharray 属性实现文字边缘动画）

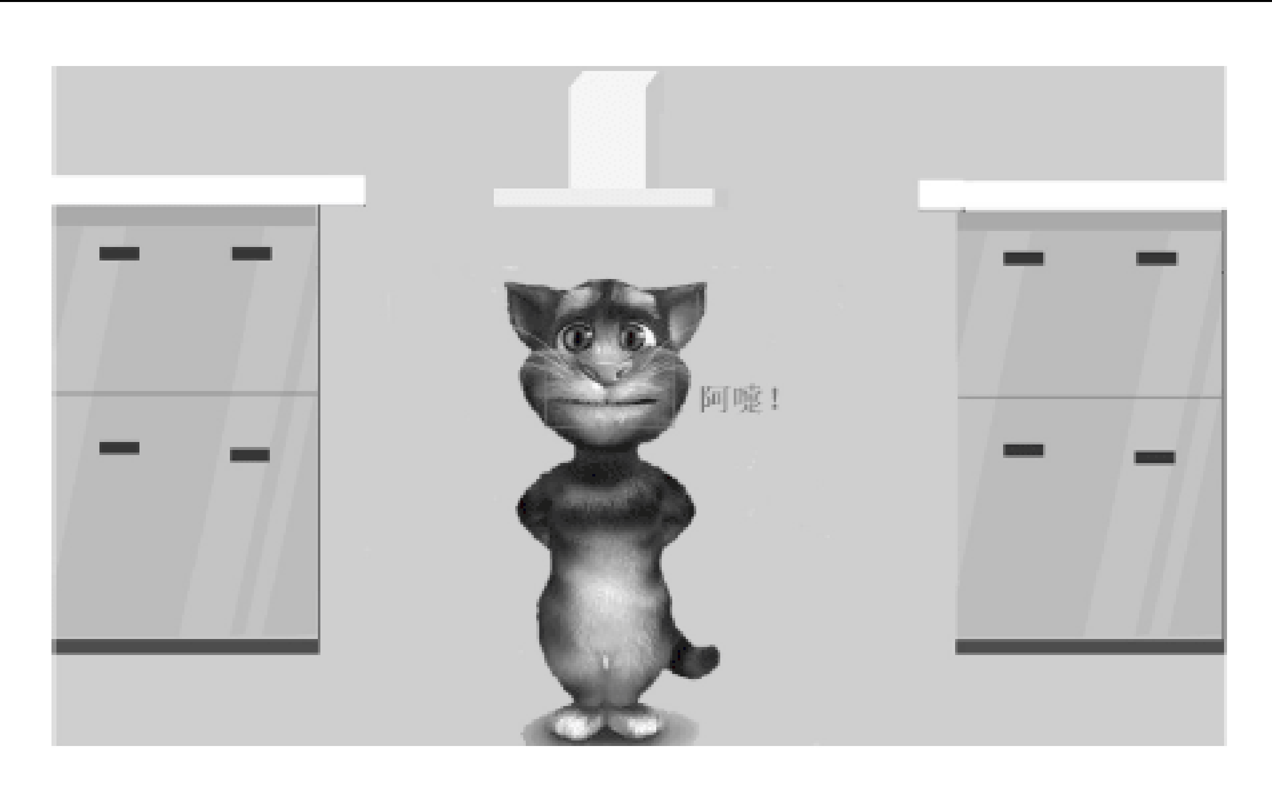


图 10.2　会说话的汤姆猫

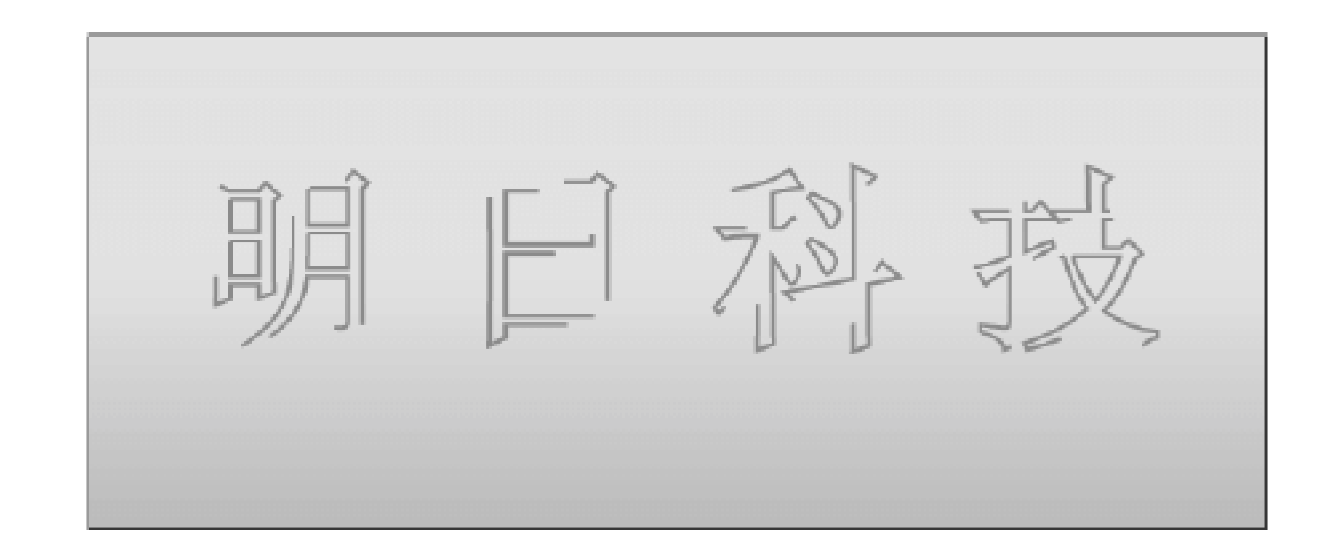


图 10.3　文字边缘镂空动画

3．恐龙爬坡动画　▷①②③④⑤⑥

通过 CSS3 可以实现一些简单路径的动画，而一些复杂的、指定路径的动画则需要通过<svg>实现。如图 10.4 所示为<svg>实现的小恐龙爬山坡动画。

（提示：要使元素按指定路径移动，可以通过<animate>或<animateMotion>标签实现）

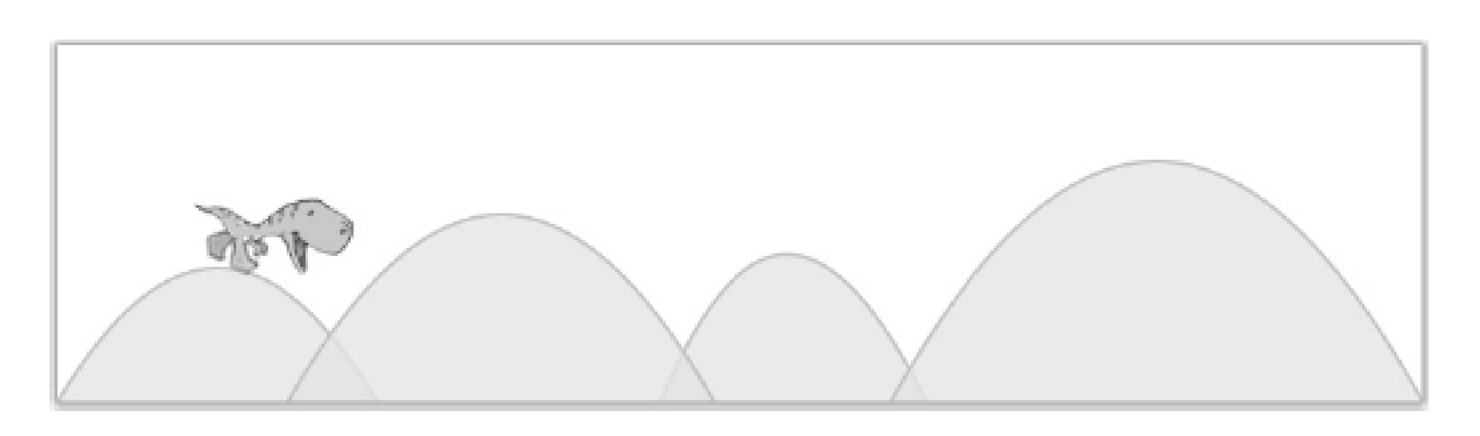


图 10.4　恐龙爬坡动画

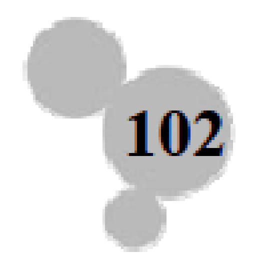
4．悬挂的日历　▷①②③④⑤⑥

编写程序，实现桌面日历效果，如图 10.5 所示。

（提示：HTML5 中新增了<fieldset>标签和<legend>标签，其中<fieldset>标签用于将相关元素组合在一起，<legend>标签是<fieldset>标签的标题，本任务将用到这两个标签）



图 10.5　悬挂的日历



5. 气温变化图　　▷①②③④⑤⑥

实现如图 10.6 所示的一周气温变化及空气质量情况图。

（提示：<progress>和<meter>标签均可实现进度条，前者注重进程，后者注重度量，本例使用<meter>标签实现）

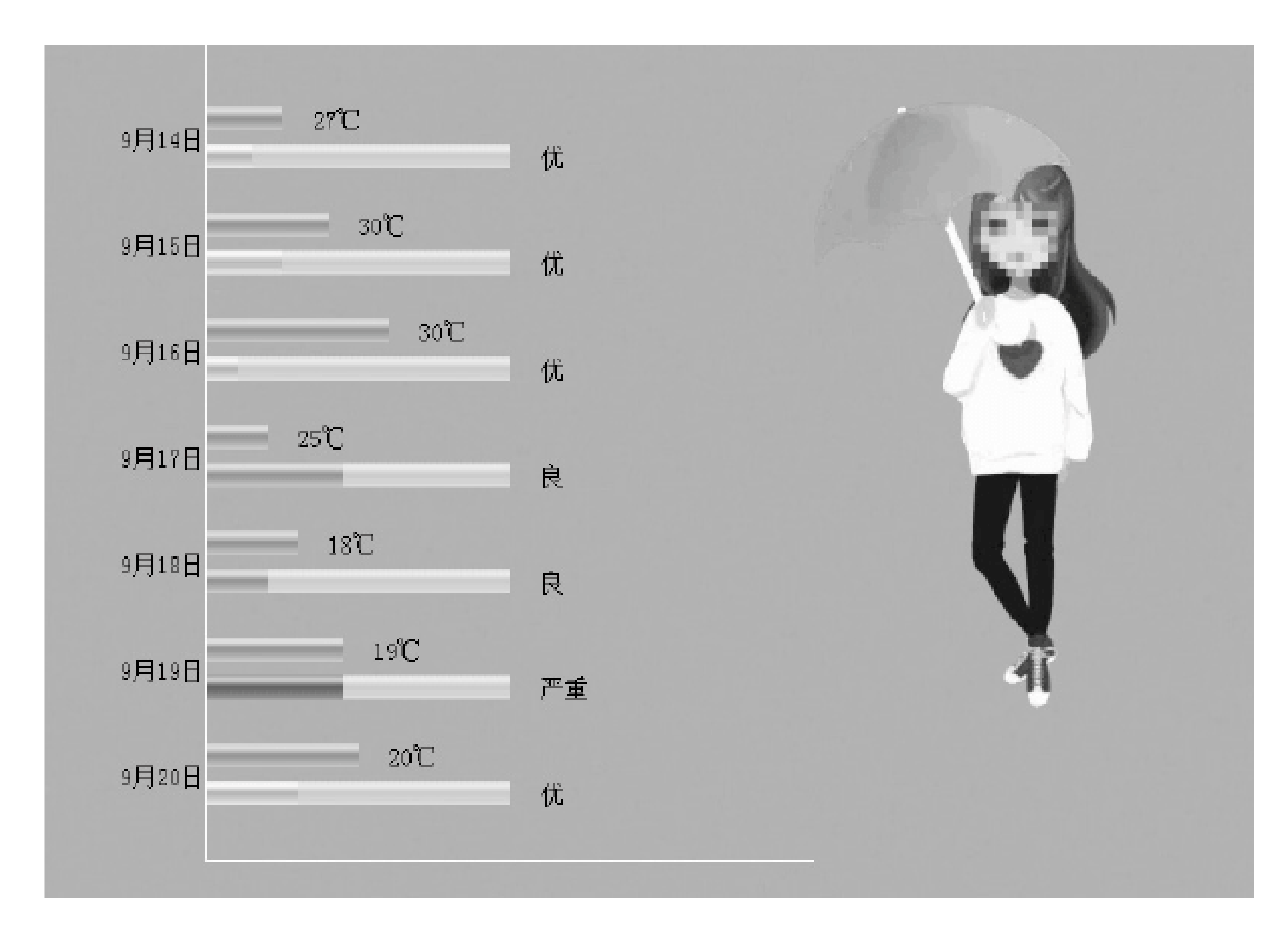


图 10.6　气温变化图

6. 网购商城商品跳转页面　　▷①②③④⑤⑥

实现一个网购商城的商品列表页面，单击左侧导航菜单，右侧页面可自动跳转至对应楼层的商品，如图 10.7 所示。注意使用语义化标签（如<nav>标签、<aside>标签等）替代部分<div>标签。

（提示：通过超链接标签<a>可实现页内跳转）



图 10.7　商品列表中楼层跳转页面

训练二：实战能力强化训练

7. 商品订购信息 ▷①②③④⑤⑥

使用 HTML5 中的表单，实现一个商品信息订购页面，效果如图 10.8 所示。

（提示：设置 input range 元素样式时，可通过 input[type=range]::-webkit-slider-thumb 语句选中滑块）



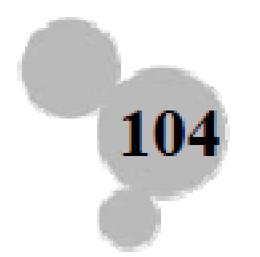
图 10.8 商品订购信息

8. 实现个人博客主页 ▷①②③④⑤⑥

使用 HTML5 新增的结构元素及语义化元素，实现一个博客页面，效果如图 10.9 所示。



图 10.9 个人博客主页



9. 将图片裁切为不规则图形 ▷①②③④⑤⑥

网页上的图形并不都是规则图形，还有很多不规则图形。这些不规则图形会使人耳目一新，记忆深刻。将元素裁剪为不规则图形可通过 CSS3 中的 clip-path 属性实现。

编写程序，通过选择页面上方的选项卡，可将图片裁切为菱形、五边形、五角星、左箭头、右箭头、对话框、错号等不规则形状，其中菱形效果如图 10.10 所示。



图 10.10　将图片裁切为不规则图形



第 11 章　JavaScript 基础

本章训练任务对应核心技术分册第 11 章“JavaScript 基础”内容。

重点练习内容：

1．JavaScript的数据类型。
2．运算符的应用。
3．流程控制语句的使用。
4．Window对象的使用。
5．JavaScript常用事件的应用。

应用技能拓展学习

1．函数的定义和调用

JavaScript 中可使用 function 语句来定义一个函数。语法格式如下：

```
function 函数名([参数 1, 参数 2,…]){
    语句
    [return 返回值]
}
```

☑ 函数名：必选项，用于指定函数的名称。同一个页面中，函数名必须具有唯一性，且区分大小写。
☑ 参数：可选项，用于指定参数列表。当使用多个参数时，参数间以逗号进行分隔。一个函数最多可以有 255 个参数。
☑ 语句：必选项，表示函数体，即用于实现函数功能的语句。
☑ 返回值：可选项，用于返回函数值。返回值可以是任意表达式、变量或常量。

例如，定义一个不带参数的函数 cal()，在函数体中输出两个数值型变量的乘积，代码如下：

```
<script type="text/javascript">
    function cal(){                    //定义函数名称为 cal
        var a=5;                       //定义函数体
        var b=6;
        alert(a*b);
    }
</script>
```

例如，定义一个用于计算商品金额的函数 account()，该函数有两个参数，用于指定商品单价和数量，返回值为计算后的金额，代码如下：

```
<script type="text/javascript">
    function account(price,number){          //定义含有两个参数的函数
        var sum=price*number;                //计算金额
        return sum;                          //返回计算后的金额
    }
</script>
```

函数定义后并不会自动执行，还需要在特定的位置调用它。函数调用语句通常包含函数名称以及各个参数的具体值。

（1）函数的简单调用

直接调用函数的语法如下：

```
函数名(传递给函数的参数 1,传递给函数的参数 2, ……);
```

例如，定义一个函数 cal()，在函数体中输出两个数值型变量的乘积，然后通过调用这个函数输出结果，代码如下：

```
<script type="text/javascript">
    function cal(){                      //定义函数，名称为 cal
        var a=5;                         //定义函数体
        var b=6;
        alert(a*b);
    }
    cal();                               //调用函数
</script>
```

运行结果：30

（2）在事件响应中调用函数

用户单击某个按钮或复选框时将触发事件，通过编写程序对事件做出反应的行为称为响应事件。在 JavaScript 语言中，将函数与事件相关联，就完成了响应事件的过程。

例如，单击“提交”按钮时弹出对话框，代码如下：

```
<script type="text/javascript">
    function test(){                     //定义函数
        alert("我喜欢 JavaScript ");     //定义函数体
    }
</script>
<button onClick="test();">提交</button>
```

（3）通过链接调用函数

函数除了可在响应事件中被调用之外，还可以在链接中被调用。在<a>标签的 href 属性中使用“javascript:函数名()”格式来调用函数，当用户单击这个链接时，相关函数将被执行。

例如，单击“单击链接”后弹出相应对话框，代码如下：

```
<script type="text/javascript">
```

```
    function test(){                        //定义函数
        alert("我喜欢 JavaScript");          //定义函数体
    }
</script>
<a href="javascript:test();">单击链接</a>       <!--在链接中调用自定义函数-->
```

2. for 语句

for 循环语句又被称为计次循环语句，一般用于循环次数已知的情况下，在 JavaScript 中应用比较广泛。for 循环语句的语法格式如下：

```
for(初始化表达式;条件表达式;迭代表达式){
    语句
}
```

- ☑ 初始化表达式：初始化语句，用来对循环变量进行初始化赋值。
- ☑ 条件表达式：循环条件，通常是一个包含比较运算符的表达式，用来限定循环变量的边限。如果循环变量超过了该边限，则停止该循环语句的执行。
- ☑ 迭代表达式：用来改变循环变量，从而控制循环次数。通常是对循环变量进行增大或减小操作。
- ☑ 语句：用来指定循环体，在循环条件的结果为 true 时，重复执行。

例如，应用 for 语句输出 1～10 这 10 个数字，代码如下：

```
<script type="text/javascript">
    for(var i=1;i<=10;i++){
        document.write(i+"\n");
    }
</script>
```

运行结果：1 2 3 4 5 6 7 8 9 10

3. 数组简介

数组是 JavaScript 中的一种复合数据类型。变量中保存的是单个数据，数组中保存的则是多个数据集合。数组是数组元素的集合，每个数组元素都有一个索引号（数组的下标），通过索引号可以方便地引用数组元素。数组的下标从 0 开始算起，第一个元素的下标是 0，第二个元素的下标是 1，以此类推。

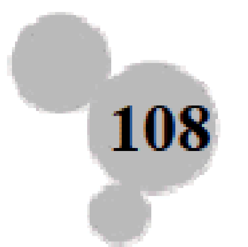
JavaScript 中主要有以下 4 种定义数组的方法。

（1）定义空数组

使用不带参数的构造函数可以定义一个空数组，后期再向数组中添加数组元素。定义空数组的语法如下：

```
arrayObject = new Array()
```

参数 arrayObject 为必选参数，表示新创建的数组对象的名称。



例如，创建一个空数组，然后向该数组中添加数组元素，代码如下：

```
<script type="text/javascript">
    var arr = new Array();
    arr[0] = "HTML";
    arr[1] = "CSS";
    arr[2] = "JavaScript";
</script>
```

（2）指定数组长度

定义数组的同时可以指定数组元素的个数。语法如下：

```
arrayObject = new Array(size)
```

☑ arrayObject：必选项，表示新创建的数组对象的名称。
☑ size：数组的长度。由于数组的下标是从零开始的，因此数组元素的下标范围是 0～size-1。

例如，创建一个元素个数为 3 的数组，并向该数组中存入数据，代码如下：

```
<script type="text/javascript">
    var arr = new Array(3);
    arr[0] = 1;
    arr[1] = 2;
    arr[2] = 3;
</script>
```

（3）指定数组元素

定义数组的同时可以直接给出数组元素的值，此时数组的长度就是括号中给出的数组元素的个数。语法如下：

```
arrayObject = new Array(element1, element2, element3, ...)
```

☑ arrayObject：必选项，表示新创建的数组对象的名称。
☑ element：存入数组中的元素。使用该定义方式时，必须指定一个以上元素。

例如，创建数组对象的同时向该对象中存入数组元素，代码如下：

```
<script type="text/javascript">
    var arr = new Array(123, "JavaScript", true);
</script>
```

（4）直接定义数组

还有一种定义数组的方式，这种方式不需要使用构造函数，直接将数组元素放在一个中括号中，元素与元素之间用逗号分隔。语法如下：

```
arrayObject = [element1, element2, element3, ...]
```

☑ arrayObject：必选项，表示新创建的数组对象名。
☑ element：存入数组中的元素。使用该定义方式时，必须指定一个以上元素。

例如，直接定义一个含有 3 个元素的数组，代码如下：

```
<script type="text/javascript">
    var arr = [123, "JavaScript", true];
</script>
```

下面介绍与数组相关的一些常用属性和方法。

（1）数组的 length 属性

length 属性用于返回数组的长度。语法如下：

```
arrayObject.length
```

参数 arrayObject 为定义的数组名称。

例如，输出已创建的数组对象的长度。代码如下：

```
<script type="text/javascript">
    var arr = ["HTML", "CSS", "JavaScript"];
    document.write(arr.length);
</script>
```

运行结果：3

（2）数组的 sort()方法

sort()方法用于对数组中的元素排序。语法如下：

```
arrayObject.sort(sortby)
```

☑ arrayObject：必选项，为定义的数组名称。

☑ sortby：可选项。规定排序的顺序，必须是函数。

📖**说明**：如果调用 sort()方法时没有使用参数，将按字母顺序对数组中的元素进行排序，也就是按照字符的编码顺序进行排序。如果想按照其他标准进行排序，就需要提供比较函数。

例如，将数组中的元素按字符编码顺序进行排序。代码如下：

```
<script type="text/javascript">
    var arr=new Array("JavaScript","HTML","CSS");
    arr.sort();
    document.write(arr);
</script>
```

运行结果：CSS,HTML,JavaScript

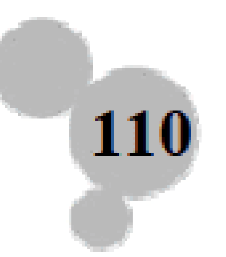
如果想要将数组元素按照其他方法进行排序，就需要指定 sort()方法的参数。该参数通常是一个比较函数，该函数应该有两个参数（假设为 a 和 b）。在对元素进行排序时，每次比较两个元素都会执行比较函数，并将这两个元素作为参数传递给比较函数。其返回值有以下两种情况：

☑ 如果返回值大于 0，则交换两个元素的位置。

☑ 如果返回值小于等于 0，则不进行任何操作。

例如，定义一个包含 4 个元素的数组，将数组中的元素按从小到大的顺序进行输出。代码如下：

```
<script type="text/javascript">
    var arr=new Array(9,6,10,5);
    function ascOrder(x,y){
```



```
            if(x>y){
                return 1;
            }else{
                return -1;
            }
        }
        arr.sort(ascOrder);
        document.write(arr);
</script>
```

运行结果：5,6,9,10

4．toFixed()方法

toFixed()方法是 Number 对象的一个方法，用于将数值四舍五入为指定小数位数的数字。语法如下：

```
number.toFixed(x)
```

参数 x 为必选参数。用于规定小数的位数，是 0～20 的值，包括 0 和 20，有些实现可以支持更大的数值范围。如果省略了该参数，将用 0 代替。

例如，将数值 5.623 保留一位小数并四舍五入。代码如下：

```
<script type="text/javascript">
    var num=5.623;
    document.write(num.toFixed(1));
</script>
```

运行结果：5.6

5．Math 对象简介

Math 对象提供了大量的数学常量和数学函数。使用 Math 对象时，不能使用 new 关键字创建对象实例，而应直接使用“对象名.成员”的格式访问其属性或方法。

（1）floor()方法

floor()方法用于返回小于等于 x 的最大整数。如果传递的参数是一个整数，则该值不变。语法如下：

```
Math.floor(x)
```

参数 x 为必选参数，表示用于操作的数字。

例如，对不同的数应用 floor()方法。代码如下：

```
<script type="text/javascript">
    var a=Math.floor(2.6);
    var b=Math.floor(2.3);
    var c=Math.floor(6);
    var d=Math.floor(6.1);
    var e=Math.floor(-6.1);
    var f=Math.floor(-6.9);
```

```
    document.write(a+"<br>");
    document.write(b+"<br>");
    document.write(c+"<br>");
    document.write(d+"<br>");
    document.write(e+"<br>");
    document.write(f);
</script>
```

运行结果：2
2
6
6
−7
−7

（2）abs()方法

abs()方法用于返回一个数字的绝对值。语法如下：

```
Math.abs(x)
```

参数 x 为必选参数，表示用于操作的数字。

例如，对不同的数字应用 abs()方法。代码如下：

```
<script type="text/javascript">
    var a=Math.abs(2.6);
    var b=Math.abs(-3.6);
    document.write(a+"<br>");
    document.write(b);
</script>
```

运行结果：2.6
3.6

6．Number()函数

Number()函数是一个 JavaScript 全局函数，用于将对象的值转换为数字。如果对象的值无法转换为数字，那么 Number()函数将返回 NaN。语法格式如下：

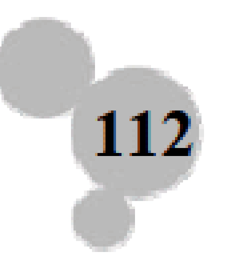
```
Number(object)
```

参数 object 为可选参数，表示一个 JavaScript 对象。如果没有提供参数，则返回 0。

📖**说明**：如果参数是 Date 对象，Number()函数将返回从 1970 年 1 月 1 日至今的毫秒数。

例如，将不同的对象转换为数字。代码如下：

```
<script type="text/javascript">
    var bool = true;
    var date = new Date();
    var str = "666";
```



```
    document.write(Number(bool)+"<br>");
    document.write(Number(date)+"<br>");
    document.write(Number(str));
</script>
```

运行结果：1
1561957737369
666

7. eval()函数

eval()函数是一个 JavaScript 全局函数，用于计算 JavaScript 字符串，并把它作为脚本代码来执行。语法格式如下：

```
eval(string)
```

参数 string 为必选参数，表示要计算的字符串，其中含有要计算的 JavaScript 表达式或要执行的语句。

例如，应用 eval()方法执行 JavaScript 语句或表达式。代码如下：

```
<script type="text/javascript">
    eval("x=10;y=20;document.write(x+y)");
    document.write("<br>" + eval("2*3"));
    document.write("<br>" + eval(x+26));
</script>
```

运行结果：30
6
36

8. parseInt()函数

parseInt()函数可解析一个字符串，并返回一个整数。语法如下：

```
parseInt(string, radix)
```

☑ string：要被解析的字符串。
☑ radix：可选参数，表示要解析的数字的基数，该值范围是 2～36。

注意：如果字符串的第一个字符不能被转换为数字，那么 parseInt()函数将返回 NaN。

例如，将字符串“123abc”转换为整数。代码如下：

```
<script type="text/javascript">
    var str="123abc";                    //定义字符串
    document.write(parseInt(str));
</script>
```

运行结果：123

9. Date 对象

在 JavaScript 中，使用 Date 对象可以操作日期和时间。

（1）创建 Date 对象

要使用 Date 对象操作日期和时间，首先要使用 new 运算符创建 Date 对象。创建一个不带参数的 Date 对象的语法格式如下：

```
dateObj = new Date()
```

参数 dateObj 表示要赋值为 Date 对象的变量名。

创建一个指定日期 Date 对象的语法格式如下：

```
dateObj = new Date(year, month, date[, hours[, minutes[, seconds[,ms]]]])
```

各参数的说明如表 11.1 所示。

表 11.1 Date 对象的参数说明

参　数	说　明
dateObj	必选项，表示要赋值为 Date 对象的变量名
year	必选项，表示完整的年份，比如 1976（而不是 76）
month	必选项，表示月份，是 0～11 的整数（1 月至 12 月）
date	必选项，表示日期，是 1～31 的整数
hours	可选项，表示小时，是 0～23 的整数。如果提供了 minutes 则必须给出
minutes	可选项，表示分钟，是 0～59 的整数。如果提供了 seconds 则必须给出
seconds	可选项，表示秒钟，是 0～59 的整数。如果提供了 ms 则必须给出
ms	可选项，表示毫秒，是 0～999 的整数

例如，在页面中输出当前的日期和时间，代码如下：

```
<script type="text/javascript">
    var newDate=new Date();
    document.write(newDate);
</script>
```

运行结果：Thu Sep 12 2019 09:43:07 GMT+0800 (中国标准时间)

（2）getTime()方法

Date 对象提供的 getTime()方法可以返回指定日期和时间距 1970 年 1 月 1 日之间的毫秒数。语法格式如下：

```
dateObj.getTime()
```

参数 dateObj 为创建的 Date 对象名。

例如，输出指定日期和时间距 1970 年 1 月 1 日之间的毫秒数，代码如下：

```
<script type="text/javascript">
    var now = new Date(2019,6,19,10,26,36);
```



```
    document.write(now.getTime());
</script>
```

运行结果：1563503196000

10．setTimeout()方法

setTimeout()方法是 Window 对象中的方法，用于在指定毫秒数后调用函数或计算表达式。语法格式如下：

```
setTimeout(要执行的代码或函数，等待的毫秒数)
```

例如，经过 3 秒钟执行 JavaScript 代码，在页面中弹出一个对话框。代码如下：

```
<script type="text/javascript">
    setTimeout("alert('对不起，让您久等了！ ')", 3000 )
</script>
```

例如，经过 3 秒钟执行 JavaScript 函数，使页面中指定的文本消失。代码如下：

```
<p id="content">三秒钟后消失!</p>
<script type="text/javascript">
    setTimeout("changeState()",3000 );
    function changeState(){
        var content=document.getElementById('content');
        content.innerHTML="";
    }
</script>
```

11．事件对象简介

IE 浏览器中，事件对象指的是 window 对象的属性 event，只在事件发生的时候被访问，事件处理完之后就消失。在标准的 DOM 浏览器中，event 必须作为唯一的参数传给事件处理函数。为了实现兼容性，通常采用下面的方法：

```
<script type="text/javascript">
    function someHandle(e) {
        //处理兼容性，获得事件对象
        var e = e || window.event;
    }
</script>
```

在标准 DOM 浏览器中，发生事件的元素通过 event 对象的 target 属性获取。例如，单击“测试”按钮时获取触发事件元素的节点名称，代码如下：

```
<script type="text/javascript">
    function test(e) {
        var e = e || window.event;
        alert(e.target.nodeName);
```

```
    }
</script>
<button onclick="test()">测试</button>
```

运行结果：BUTTON

（1）clientX 和 clientY 事件属性

clientX 事件属性用于返回事件被触发时鼠标指针相对于浏览器页面（或客户区）的水平坐标，clientY 事件属性用于返回事件被触发时鼠标指针相对于浏览器页面（或客户区）的垂直坐标。这里，客户区表示当前窗口。

clientX 属性的语法格式如下：

```
event.clientX
```

clientY 属性的语法格式如下：

```
event.clientY
```

例如，单击页面时获取鼠标指针的坐标，代码如下：

```
<script type="text/javascript">
    document.onmousedown=function (e) {
        var x=e.clientX;
        var y=e.clientY;
        alert("X 坐标："+x+"，Y 坐标："+y);
    }
</script>
```

运行结果如图 11.1 所示。



图 11.1　获取鼠标指针的坐标

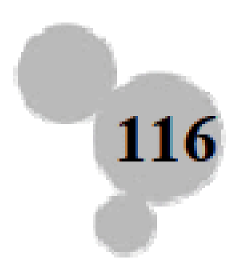
（2）dispatchEvent()方法

dispatchEvent()方法是事件对象的一个方法，用于向指定事件目标派发一个事件。

例如，向 id 为 show 的按钮派发一个单击事件。代码如下：

```
<button id="show" onclick="alert('先完成一个小目标')">提示</button>
<script type="text/javascript">
    var myEvent = new Event('click');
    document.getElementById("show").dispatchEvent(myEvent);
</script>
```

运行结果：先完成一个小目标



12．为元素绑定事件

以绑定 onclick 事件为例，JavaScript 中为元素绑定 onclick 事件的语法如下：

```
object.onclick=function(){
SomeJavaScriptCode
};
```

参数 SomeJavaScriptCode 为必选参数。用于规定该事件发生时执行的 JavaScript 代码。

例如，为“测试”按钮绑定 onclick 事件，单击按钮时弹出相应的对话框。代码如下：

```
<button id="test">测试</button>
<script type="text/javascript">
    document.getElementById("test").onclick = function () {
        alert("您单击了测试按钮");
    }
</script>
```

13．DOM 简介

DOM 是 Document Object Model（文档对象模型）的缩写。网页被加载时，浏览器会创建页面的文档对象模型。文档对象模型采用的分层结构为树形结构，以树节点方式表示文档中的各项内容。在文档对象模型中，每个对象都被称为一个节点（Node）。下面介绍一下几种节点的概念。

- ☑ 根节点：最顶层的<html>节点。
- ☑ 父节点：某节点之上的节点称为该节点的父节点（parent）。例如，<html>是<head>和<body>的父节点，<head>是<title>的父节点。
- ☑ 子节点：某节点之下的节点称为该节点的子节点（child）。例如，<head>和<body>是<html>的子节点，<title>是<head>的子节点。
- ☑ 兄弟节点：位于同一层次且拥有相同父节点的节点就是兄弟节点（sibling）。例如，<head>和<body>就是兄弟节点。

下面来看一下与文档对象相关的属性和方法。

（1）nodeName 属性

nodeName 属性可以根据节点类型返回其名称。语法格式如下：

```
node.nodeName
```

例如，获取 id 属性值为 js 的元素的节点名称，代码如下：

```
<div id="js">JavaScript</div>
<script type="text/javascript">
    alert(document.getElementById("js").nodeName);
</script>
```

运行结果：DIV

（2）value 属性

value 属性用于设置或者返回属性的值。

设置属性值的语法格式如下：

```
attribute.value=value
```

返回属性值的语法格式如下：

```
attribute.value
```

例如，单击“提交”按钮时输出文本框的值，代码如下：

```
<input type="text" id="demo" size="15">
<input type="button" value="提交" onclick="show()">
<script type="text/javascript">
    function show(){
        alert(document.getElementById("demo").value);
    }
</script>
```

运行结果如图 11.2 所示。



图 11.2　实现效果

（3）checked 属性

checked 属性用于设置或返回复选框是否应被选中。

设置 checked 属性的语法格式如下：

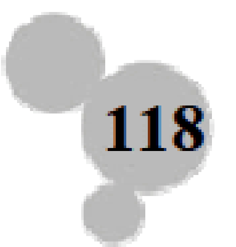
```
checkboxObject.checked=true|false
```

返回 checked 属性的语法格式如下：

```
checkboxObject.checked
```

例如，单击“选中”按钮时，设置复选框为选中状态，代码如下：

```
<input type="checkbox" id="agree">同意协议
<input type="button" value="选中" onclick="check()">
<script type="text/javascript">
    function check(){
        document.getElementById("agree").checked=true;
    }
</script>
```



运行结果如图 11.3 和图 11.4 所示。

图 11.3　原始效果　　　　图 11.4　选中复选框

（4）innerHTML 属性

innerHTML 属性用于设置或返回元素含有的 HTML 文本，不包括元素本身的开始标记和结束标记。例如，通过 innerHTML 属性设置<div>标记的内容，代码如下：

```
<div id="demo"></div>
<script type="text/javascript">
    document.getElementById("demo").innerHTML="功夫不负有心人";
</script>
```

运行结果：功夫不负有心人

（5）className 属性

className 属性用于设置或返回元素的 class 属性值。

获取 class 属性值的语法格式如下：

```
HTMLElementObject.className
```

设置 class 属性值的语法格式如下：

```
HTMLElementObject.className=classname
```

例如，获取指定 div 元素的 class 属性值，代码如下：

```
<div id="myid" class="mystyle"></div>
<script type="text/javascript">
    document.write(document.getElementById('myid').className);
</script>
```

运行结果：mystyle

（6）getElementById()方法

getElementById()方法是 DOM 中定义的查找元素方法，用于返回对拥有指定 id 的第一个对象的引用。语法如下：

```
document.getElementById(elementID)
```

参数 elementID 为必选参数，表示元素的 id 属性值。

例如，输出 id 属性值为 demo 的元素的 HTML 内容，代码如下：

```
<div id="demo">明日科技</div>
<script type="text/javascript">
    alert(document.getElementById("demo").innerHTML);
</script>
```

运行结果：明日科技

（7）getElementsByTagName()方法

getElementsByTagName()方法是 DOM 中定义的查找元素的方法，用于返回带有指定标签名的对象的集合。语法如下：

```
document.getElementsByTagName(tagname)
```

参数 tagname 为必选参数，表示需要获取元素的标签名。

例如，设置第一个 div 元素的 HTML 内容，代码如下：

```
<div></div>
<script type="text/javascript">
    document.getElementsByTagName("div")[0].innerHTML="笑书神侠倚碧鸳";
</script>
```

运行结果：笑书神侠倚碧鸳

（8）querySelector()方法

querySelector()方法用于返回文档中匹配指定 CSS 选择器的一个元素，且仅返回匹配指定选择器的第一个元素。语法如下：

```
document.querySelector(selectors)
```

参数 selectors 为必选参数，用于指定一个或多个匹配元素的 CSS 选择器。可以使用其 id、类、类型、属性、属性值等来选取元素。多个选择器之间使用逗号隔开。

例如，将页面中第一个 div 元素的内容修改为“JavaScript”，代码如下：

```
<div>HTML</div>
<div>CSS</div>
<script type="text/javascript">
    document.querySelector("div").innerHTML = "JavaScript";
</script>
```

运行结果：JavaScript
　　　　　CSS

（9）querySelectorAll()方法

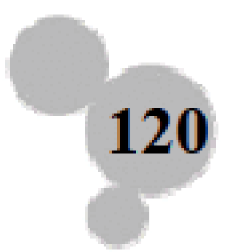
querySelectorAll()方法用于返回文档中匹配指定 CSS 选择器的所有元素。语法如下：

```
document.querySelectorAll(selectors)
```

参数 selectors 为必选参数，用于指定一个或多个匹配元素的 CSS 选择器。可以使用其 id、类、类型、属性、属性值等来选取元素。多个选择器之间使用逗号隔开。

例如，将页面中第二个 div 元素的内容修改为“JavaScript”，代码如下：

```
<div>HTML</div>
<div>CSS</div>
<script type="text/javascript">
    var div = document.querySelectorAll("div");
    div[1].innerHTML = "JavaScript";
</script>
```



（10）getElementsByClassName()方法

getElementsByClassName()方法是 DOM 中定义的用于查找元素的方法，可返回文档中所有指定类名的元素集合。语法如下：

```
document.getElementsByClassName(classname)
```

参数 classname 为必选参数，表示需要获取的元素类名，多个类名之间使用空格分隔。

例如，设置类名为 wuxia 的第一个元素的 HTML 内容，代码如下：

```
<div class="wuxia"></div>
<script type="text/javascript">
    document.getElementsByClassName("wuxia")[0].innerHTML="飞雪连天射白鹿";
</script>
```

运行结果：飞雪连天射白鹿

14．style 对象简介

style 对象是 HTML 对象的一个属性。style 对象提供了一组对应于浏览器所支持的 CSS 样式的属性（如 background、fontSize、borderColor 等）。每一个 HTML 对象都有一个 style 属性，可以使用这个属性访问 CSS 样式属性。

在 style 对象中，样式标签属性和样式属性基本上是相互对应的，两种属性的用法也基本相同。唯一的区别是样式标签属性用于设置对象的属性，而样式属性用于检索或更改对象的属性。也可以说，样式标签属性是静态属性，样式属性是动态属性。

例如，利用 style 对象改变字体的大小，代码如下：

```
<style type="text/css">
    body{
        color: green;
    }
    p{
        font-size: 12px;
    }
</style>
<body>
<p id="pid">改变字体大小</p>
<script type="text/javascript">
    document.getElementById("pid").style.fontSize = '36px';
</script>
</body>
```

（1）width 属性

width 属性可以设置或返回元素的宽度。

设置 width 属性的语法格式如下：

```
Object.style.width="auto|length|%|inherit"
```

☑ auto：默认值，表示浏览器设置的宽度。
☑ length：使用 px、cm 等单位定义的宽度。
☑ %：定义宽度为父元素的百分比。
☑ inherit：width 属性的值从父元素继承。

返回 width 属性的语法格式如下：

```
Object.style.width
```

例如，单击“改变宽度”按钮时，设置按钮的宽度为 100 像素，代码如下：

```
<script type="text/javascript">
    function changeWidth(){
        document.getElementById("but").style.width="100px";
    }
</script>
<input type="button" id="but" onclick="changeWidth()" value="改变宽度">
```

运行结果如图 11.5 和图 11.6 所示。

图 11.5　单击前

图 11.6　单击后

（2）opacity 属性

opacity 属性可以设置或返回元素的不透明度。当元素的不透明度为 1 时，表示完全不透明；当元素的不透明度为 0.5 时，表示半透明；当元素的不透明度为 0 时，表示完全透明。

设置 opacity 属性的语法格式如下：

```
object.style.opacity="number|initial|inherit"
```

☑ number：指定的不透明度。
☑ initial：设置不透明度为其默认值。
☑ inherit：从父元素继承不透明度值。

返回 opacity 属性的语法格式如下：

```
object.style.opacity
```

例如，设置 id 属性值为 js 的 div 元素的不透明度为 0.5，代码如下：

```
<style type="text/css">
    div{
        width: 100px;
        height: 50px;
        line-height: 50px;
        background: green;
    }
</style>
<div id="js">JavaScript</div>
<script type="text/javascript">
    document.getElementById("js").style.opacity=0.5;
</script>
```

运行结果如图 11.7 所示。

（3）left 属性

left 属性用于设置或返回定位元素的左部位置。

设置 left 属性的语法格式如下：

```
Object.style.left="auto|length|%|inherit"
```

- ☑ auto：默认值，表示通过浏览器计算左部的位置。
- ☑ length：使用 px、cm 等单位定义元素左边距离最近一个具有定位设置父元素的左部边缘的位置。可使用负值。
- ☑ %：定义元素左边距离最近一个具有定位设置父元素的左部边缘的百分比位置。
- ☑ inherit：left 属性的值从父元素继承。

返回 left 属性的语法格式如下：

```
Object.style.left
```

例如，设置 id 属性值为 demo 的元素到页面左端的距离为 50 像素，代码如下：

```
<div id="demo" style="position: absolute;">明天会更好</div>
<script type="text/javascript">
    document.getElementById("demo").style.left="50px";
</script>
```

运行结果如图 11.8 所示。

（4）top 属性

top 属性用于设置或返回定位元素的顶部位置。

设置 top 属性的语法格式如下：

```
Object.style.top="auto|length|%|inherit"
```

- ☑ auto：默认值，表示通过浏览器计算顶部的位置。
- ☑ length：使用 px、cm 等单位定义元素顶边距离最近一个具有定位设置父元素的顶部边缘的位置。可使用负值。
- ☑ %：定义元素顶边距离最近一个具有定位设置父元素的顶部边缘的百分比位置。
- ☑ inherit：top 属性的值从父元素继承。

返回 top 属性的语法格式如下：

```
Object.style.top
```

例如，设置 id 属性值为 demo 的元素到页面顶端的距离为 100 像素，代码如下：

```
<div id="demo" style="position: absolute;">明天会更好</div>
<script type="text/javascript">
    document.getElementById("demo").style.top="100px";
</script>
```

运行结果如图 11.9 所示。

JavaScript　　明天会更好　　明天会更好

图 11.7　设置不透明度　　图 11.8　设置 left 属性　　图 11.9　设置 top 属性

（5）display 属性

display 属性用于设置或返回元素的显示类型。该属性允许用户显示或隐藏一个元素，与 visibility 属性类似。不同的是：设置 display:none，会隐藏整个元素；设置 visibility:hidden，元素内容虽不可见，但会保留原来的位置和大小。

设置 display 属性的语法格式如下：

```
Object.style.display="value"
```

返回 display 属性的语法格式如下：

```
Object.style.display
```

display 属性的可选值及其说明如表 11.2 所示。

表 11.2　display 属性的可选值及说明

属　　性	说　　明
block	设置元素为块级元素
compact	设置元素为块级元素或内联元素，取决于上下文
inherit	display 属性值从父元素继承
inline	默认，设置元素为内联元素
inline-block	设置元素为内联盒子内的块盒子
inline-table	设置元素为内联表格（类似于<table>），表格前后没有换行符
list-item	设置元素为列表
marker	在盒子前后设置内容为标记（与:before 和:after 伪元素一起使用，否则该值与 inline 相同）
none	设置元素不会被显示
run-in	设置元素为块级或内联元素，取决于上下文
table	设置元素为块级表格（类似于<table>），表格前后带有换行符
table-caption	设置元素为表格标题（类似于<caption>）
table-cell	设置元素为表格单元格（类似于<td>和<th>）
table-column	设置元素为表格列（类似于<col>）
table-column-group	设置元素为一个或多个列的分组（类似于<colgroup>）
table-footer-group	设置元素为表格页脚行（类似于<tfoot>）
table-header-group	设置元素为表格页眉行（类似于<thead>）
table-row	设置元素为表格行（类似于<tr>）
table-row-group	设置元素为一个或多个行的分组（类似于<tbody>）

例如，单击“隐藏文本”按钮时设置文本隐藏，代码如下：

```
<script type="text/javascript">
    function hiddenText(){
        document.getElementById("demo").style.display="none";
    }
</script>
<input type="button" onclick="hiddenText()" value="隐藏文本">
<p id="demo">天生我材必有用</p>
```

运行结果如图 11.10 和图 11.11 所示。

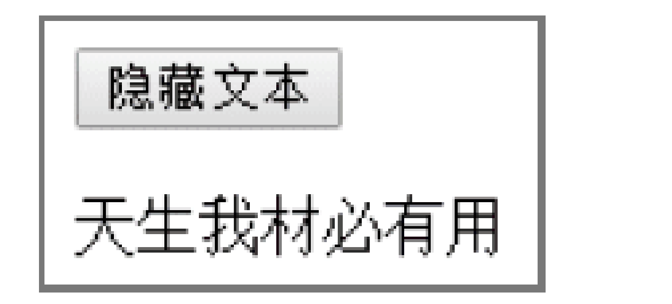


图 11.10　原始效果

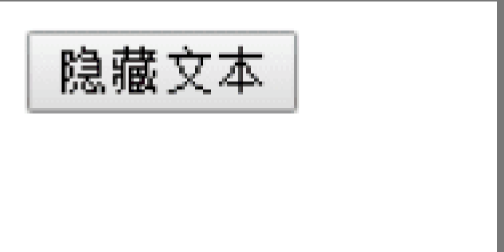


图 11.11　文本隐藏

（6）paddingLeft 属性

paddingLeft 属性可以设置或返回元素的左内边距。

设置 paddingLeft 属性的语法格式如下：

```
Object.style.paddingLeft="%|length|inherit"
```

☑ %：定义左内边距基于父元素宽度的百分比。

☑ length：使用 px、cm 等单位定义左内边距的宽度。

☑ inherit：左内边距从父元素继承。

返回 paddingLeft 属性的语法格式如下：

```
Object.style.paddingLeft
```

例如，设置 id 为 show 的元素的左内边距为 100px，代码如下：

```
<script type="text/javascript">
    document.getElementById("show").style.paddingLeft="100px";
</script>
```

15. 图像对象简介

（1）什么是图像对象

Document 对象的 images 属性的返回值是一个数组，其中的每个元素都是一个 Image 对象。Image 对象就是图像对象。HTML 文档中可能会存在多张图片，加载文档时，JavaScript 会自动创建一个 images[] 数组，数组中的元素个数是由文档中<img>标签的个数决定的。images[]数组中的每个元素都代表着文档中的一张图片。

在操作图像对象之前，首先应当确定要引用的图片。JavaScript 中有 3 种引用图片的方式，分别如下：

☑ 通过 document.images[]按编号访问图片，如 document. images[0]。

☑ 通过 document.images[imageName]按名称访问图片，如 document.images["book"]。

☑ 在支持 DOM 的浏览器中，使用 document.getElementById("imageID")定位要访问的图片。

例如，页面中有一张图片，代码如下：

```
<img name="flower" id="mypic" src="flower.png">
```

要引用该图片，可以使用 document.images[0]、document.images["flower"]或者 document.getElementById("mypic")3 种方式。

（2）width 属性和 height 属性

在图像对象中，width 属性可以设置或者返回图片的宽度，height 属性可以设置或者返回图片的高度。

width 属性的语法格式如下：

```
imageObject.width=pixels
```

height 属性的语法格式如下：

```
imageObject.height=pixels
```

例如，单击“改变图片大小”按钮时，设置图片的宽度和高度为 300 像素，代码如下：

```
<script type="text/javascript">
    function changeSize(){
        document.getElementById("mypic").width="300";
        document.getElementById("mypic").height="300";
    }
</script>
<img id="mypic" src="face.png" border="1"><br>
<input type="button" onclick="changeSize()" value="改变图片大小">
```

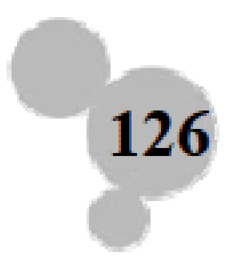
运行结果如图 11.12 和图 11.13 所示。



图 11.12　原图效果



图 11.13　大图效果



16．offsetLeft、offsetTop、offsetWidth、offsetHeight 属性

offsetLeft：元素左外边框到已设置定位属性的父元素左内边框之间的像素距离。如果父元素中没有定位元素，就返回相对于 body 左边缘的距离。

offsetTop：元素上外边框到已设置定位属性的父元素上内边框之间的像素距离。如果父元素中没有定位元素，就返回相对于 body 上边缘的距离。

offsetWidth：元素在水平方向上占用的空间大小，单位是像素。包括元素的宽度、左右内边距宽度以及左右边框的宽度。

offsetHeight：元素在垂直方向上占用的空间大小，单位是像素。包括元素的高度、上下内边距高度以及上下边框的高度。

17．scrollLeft、scrollTop 属性

scrollLeft：返回或者设置元素内容向左滚动的距离。该值是一个数字，单位是像素。

scrollTop：返回或者设置元素内容向上滚动的距离。该值是一个数字，单位是像素。

18．直接创建自定义对象的方法

直接创建自定义对象的语法格式如下：

```
var 对象名 = {属性名 1:属性值 1,属性名 2:属性值 2,属性名 3:属性值 3......}
```

直接创建自定义对象时，所有属性都放在大括号中，属性之间用逗号分隔，每个属性都由属性名和属性值两部分组成，属性名和属性值之间用冒号隔开。

例如，创建一个学生对象 student，并设置 3 个属性，分别为 name、sex 和 age，然后输出这 3 个属性的值，代码如下：

```
<script type="text/javascript">
    var student = {
        name:"张三",
        sex:"男",
        age:25
    }
    document.write("姓名："+student.name+"<br>");
    document.write("性别："+student.sex+"<br>");
    document.write("年龄："+student.age);
</script>
```

运行结果：

姓名：张三

性别：男

年龄：25

实战技能强化训练

训练一：基本功强化训练

1. 输出自动柜员机客户凭条 ▷①②③④⑤⑥

输出自动柜员机客户凭条信息，实现效果如图 11.14 所示。注意，将客户凭条中的交易日期及时间、受理银行行号、ATM 编号、交易序号、银行卡号、金额和手续费分别定义在变量中。

2. 输出《九阳神功》口诀 ▷①②③④⑤⑥

应用<pre>标签和转义字符输出《九阳神功》中的口诀，实现效果如图 11.15 所示。

自动柜员机客户凭条

交易日期及时间 2019-10-10 12:26:36	受理银行行号 0313
ATM编号 19060236	交易序号 002696
卡号 621483266569369****	
金额 300.00	手续费 0.00
转入卡号/账号	

图 11.14　输出自动柜员机客户凭条

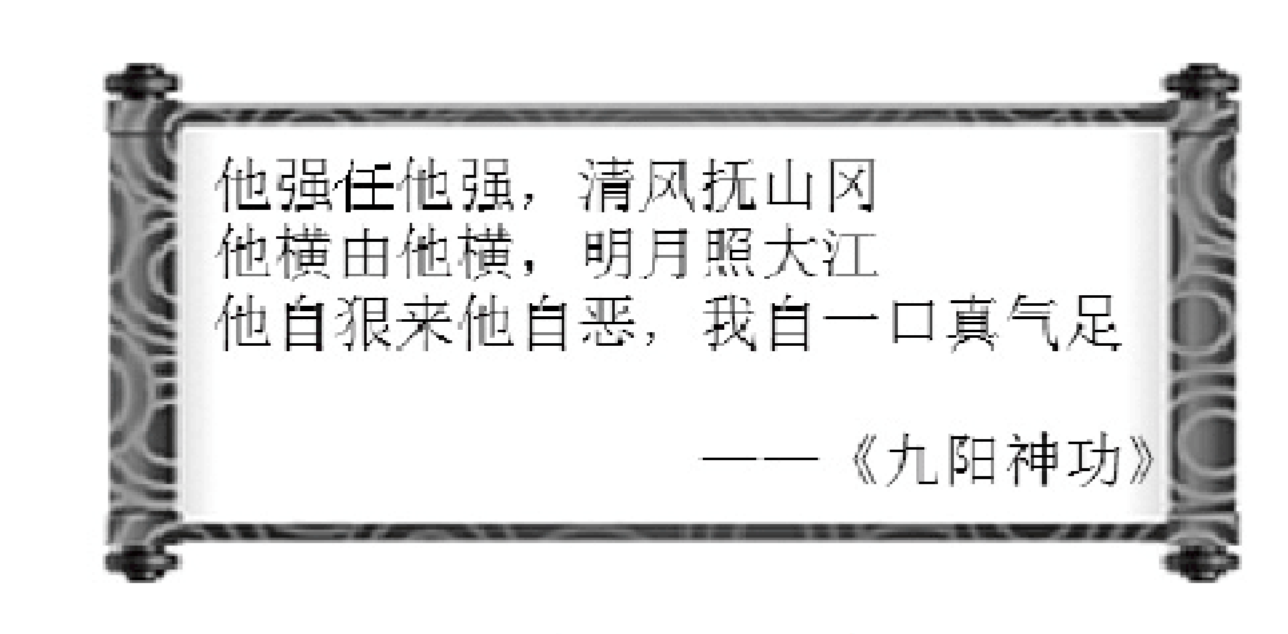


图 11.15　输出《九阳神功》口诀

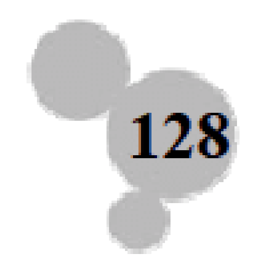
3. 计算存款本息合计 ▷①②③④⑤⑥

假设某银行定期存款 3 年的年利率为 2.75%，某客户的存款金额为 10000 元，计算该客户存款 3 年后的本息合计是多少，实现效果如图 11.16 所示。

（提示：本息合计=存款金额+存款金额*年利率*存款期限）

存款计算	
存款金额(元)	10000
存款期限(年)	3
年利率(%)	2.75
本息合计(元)	10825

图 11.16　计算存款本息合计



4．计算身体质量指数 ▷①②③④⑤⑥

身体质量指数（BMI）是目前国际上常用来衡量人体胖瘦程度以及是否健康的一个标准。BMI 值低于 18.4 表示身体偏瘦，BMI 值在 18.5 到 23.9 之间表示体重正常，BMI 值在 24 到 27.9 之间表示体重过重，BMI 值高于 28 表示身体肥胖。

试着编写一个计算 BMI 的程序。在文本框中输入身高和体重，通过计算 BMI 值判断身体状态，实现效果如图 11.17 和图 11.18 所示。



图 11.17　BMI 页面效果

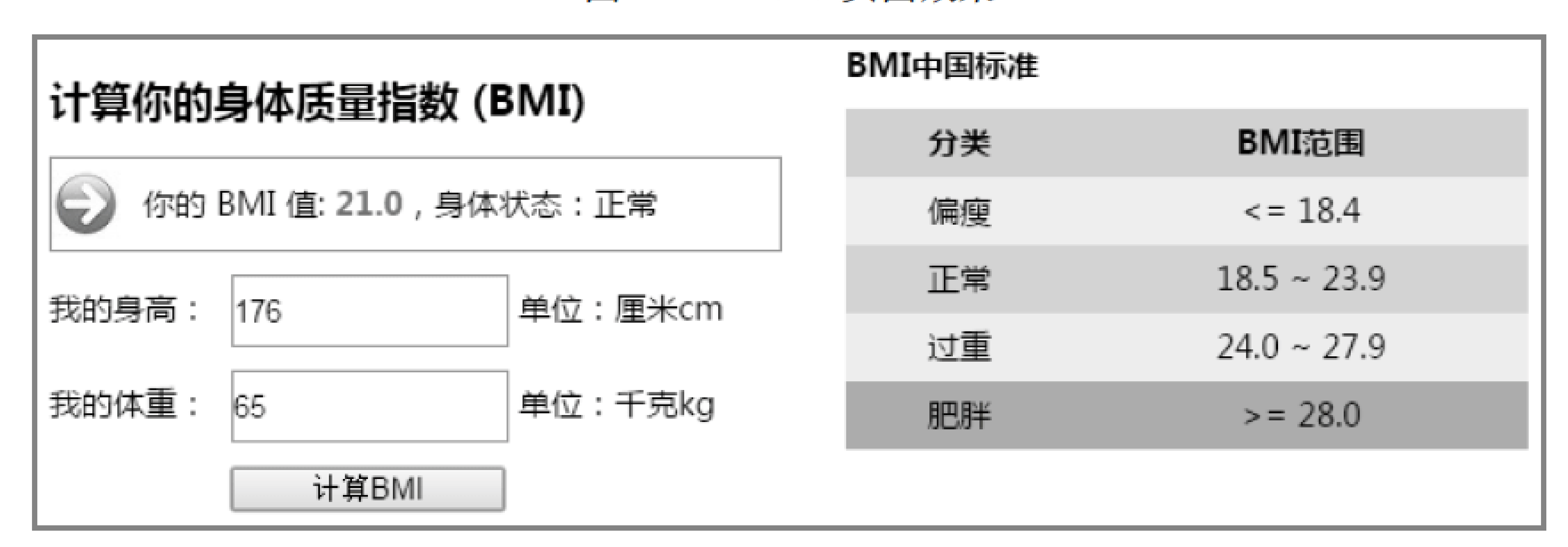


图 11.18　计算 BIM 结果

5．查看你是什么星座 ▷①②③④⑤⑥

试着编写一个查看星座的程序。将用户输入的出生日期与十二星座起止日期进行比对，判断用户属于哪个星座。在文本框中输入出生月份和出生日期，单击“查看结果”按钮可查看所属星座，实现效果如图 11.19 和图 11.20 所示。

（提示：在 if 语句中对输入的月份、日期和各个星座对应的月份和日期进行比较）

图 11.19　查看星座（1）

图 11.20　查看星座（2）

6. 输出由“*”组成的空心菱形　　▷①②③④⑤⑥

编写程序，输出由“*”组成的空心菱形图案，效果如图 11.21 所示。

（提示：在 for 语句中嵌套使用 for 语句）

7. 商品抢购倒计时　　▷①②③④⑤⑥

各大购物网站在双十一时都会推出商品抢购活动，用户在规定的时间内才可以抢购或秒杀到某些商品。编写程序，模拟购物网站中商品抢购以及商品秒杀倒计时的功能，实现效果如图 11.22 和图 11.23 所示。

（提示：使用“两个时间相隔的毫秒数/(24*60*60*1000)”，并应用 floor()方法获得相隔的天数，然后使用“两个时间相隔的毫秒数%(24*60*60*1000)/(60*60*1000)）”，并应用 floor()方法获得剩余的小时数，以此类推，根据相同的方法获得剩余的分钟数和秒数）

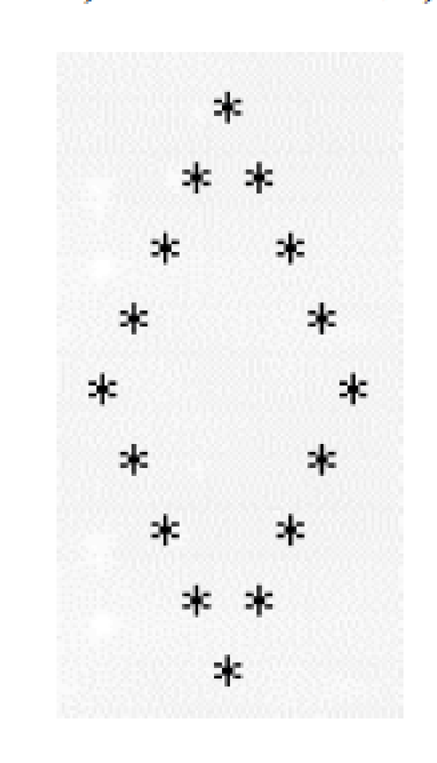


图 11.21　输出空心菱形

商品抢购即将开始
距开始 00:15:56

图 11.22　商品抢购倒计时

商品正在秒杀
距结束 00:23:56

图 11.23　商品秒杀倒计时

8. 播放歌曲权限设置　　▷①②③④⑤⑥

模拟音乐网站中不登录就无法播放歌曲的权限设置及弹出提示功能。单击歌曲对应的播放按钮，将弹出“只有会员才能播放歌曲，请登录！”提示对话框，效果如图 11.24 所示。

序号	歌曲名称	主唱	歌曲简介	下载	在线试听
01	你的样子	林志炫	⋯	↓	▷
02	只要有你	孙楠/那英	⋯	↓	▷
03	宽容	张信哲	⋯	↓	▷

此网页显示：
只有会员才能播放歌曲，请登录！
禁止此页再显示对话框。
确定

图 11.24　播放歌曲权限

9．计算从出生到现在度过的时间　　▷①②③④⑤⑥

计算自己从出生到现在度过了多长时间。在 3 个文本框中分别输入出生年份、出生月份和出生日期，输入完毕之后单击“计算”按钮，下方文本框中将显示用户到当前为止已度过的时间。实现效果如图 11.25 所示。

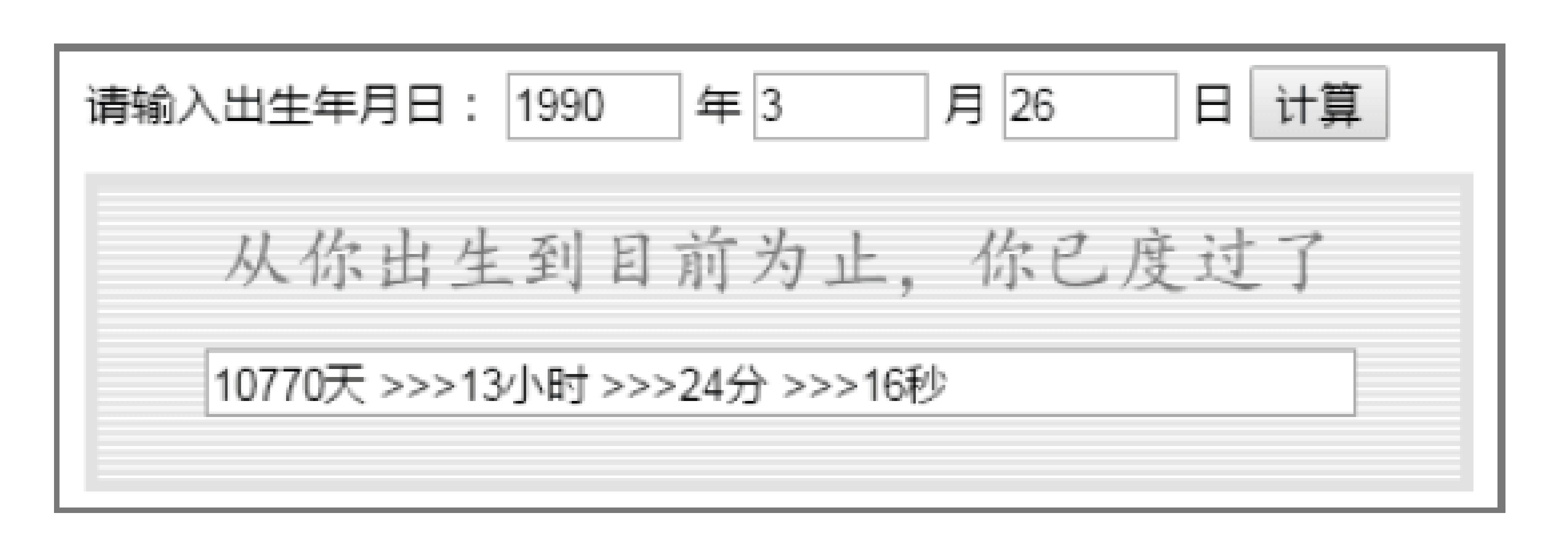


图 11.25　计算从出生到现在度过的时间

10．输出 2018 年内地电影票房排行榜　　▷①②③④⑤⑥

将2018年内地电影票房排行榜前十名的影片名称和票房定义在数组中，按影片票房进行降序排序，并将排序后的影片排名、影片名称和票房输出在页面中。实现效果如图 11.26 所示。

排名	电影名称	票房
1	红海行动	36.5
2	唐人街探案2	33.9
3	我不是药神	30.9
4	西虹市首富	25.4
5	复仇者联盟3：无限战争	23.9
6	捉妖记2	22.3
7	海王	19.5
8	毒液：致命守护者	18.7
9	侏罗纪世界2	16.9
10	头号玩家	13.9

图 11.26　2018 年内地电影票房排行榜

训练二：实战能力强化训练

11．灯泡点亮与熄灭　　▷①②③④⑤⑥

设计一个点亮灯泡和熄灭灯泡的动画效果。当鼠标光标在灯泡上按下时，灯泡被点亮；当鼠标光标在灯泡上松开时，灯泡被熄灭，效果如图 11.27 和图 11.28 所示。

12．抽屉风格的滑出菜单　▷①②③④⑤⑥

实现抽屉风格的滑出菜单。在页面中输出一个竖向的导航菜单，当鼠标移到某个菜单项上时，该菜单项会向右滑出；当鼠标移出菜单项时，该菜单项会恢复为初始状态。实现效果如图 11.29 和图 11.30 所示。

（提示：应用 event 对象的 target 属性获取发生事件的元素）



图 11.27　点亮灯泡　　图 11.28　熄灭灯泡　　图 11.29　初始菜单　　图 11.30　抽屉滑出效果

13．切换表情图片　▷①②③④⑤⑥

定义 5 个表示不同表情的单选按钮，通过单击不同的单选按钮实现表情图片的切换，效果如图 11.31 和图 11.32 所示。

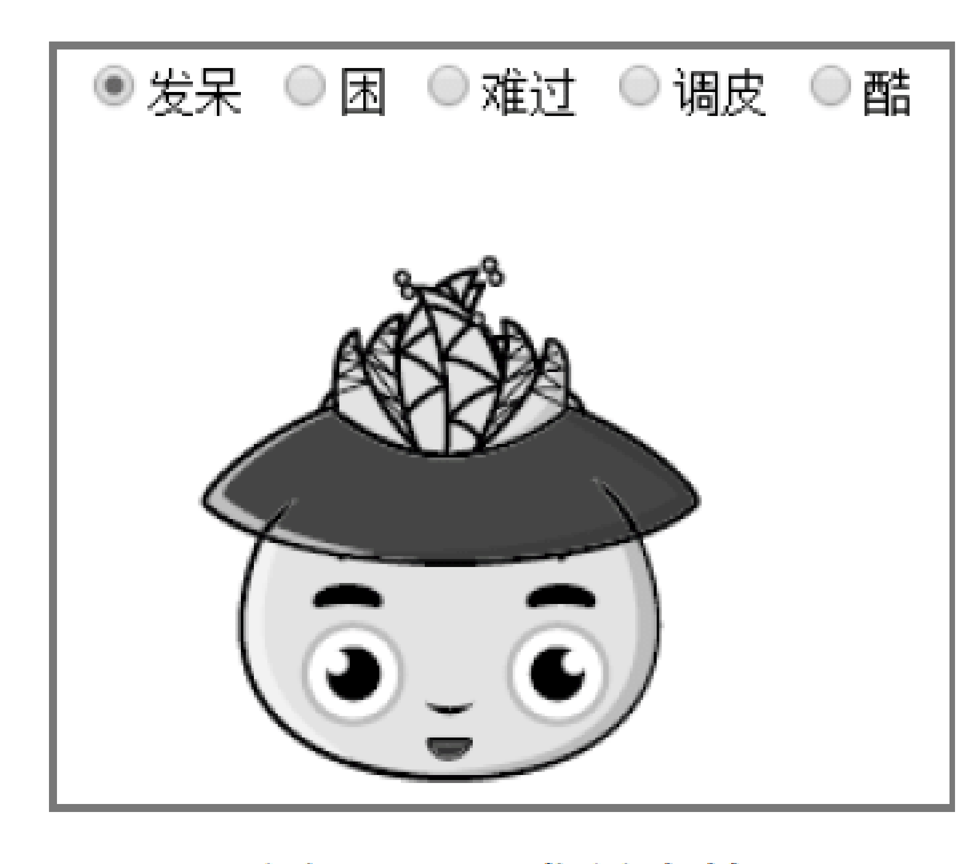


图 11.31　发呆表情

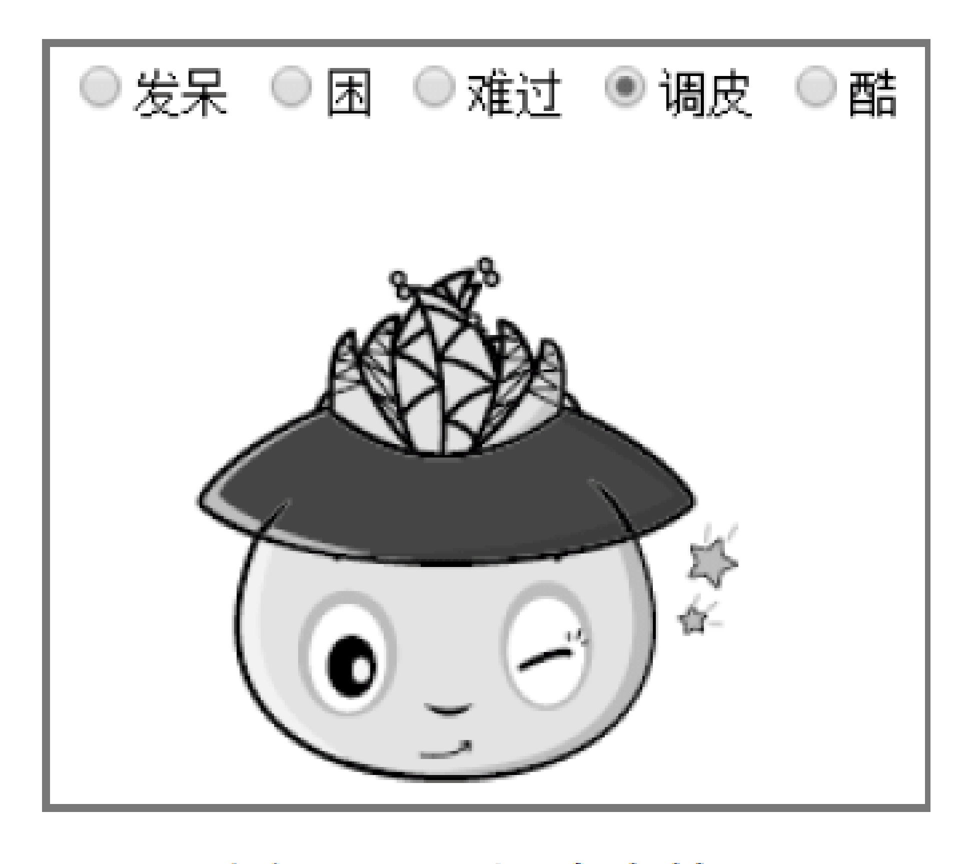
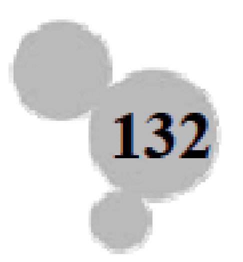


图 11.32　调皮表情

14．为图片添加和移除模糊效果　▷①②③④⑤⑥

通过改变图片的不透明度，实现图片的模糊效果。当鼠标移入图片时，改变图片的不透明度；当鼠标移出图片时，恢复图片的初始清晰度，效果如图 11.33 和图 11.34 所示。

（提示：在事件处理函数中应用 style 对象的 opacity 属性设置元素的不透明度）



15．简单计算器 ▷①②③④⑤⑥

制作一个简单的计算器。通过该计算器可以对数值进行加、减、乘、除等运算，实现效果如图 11.35 和图 11.36 所示。

（提示：在单击事件的处理函数中应用 eval()方法计算运算结果）

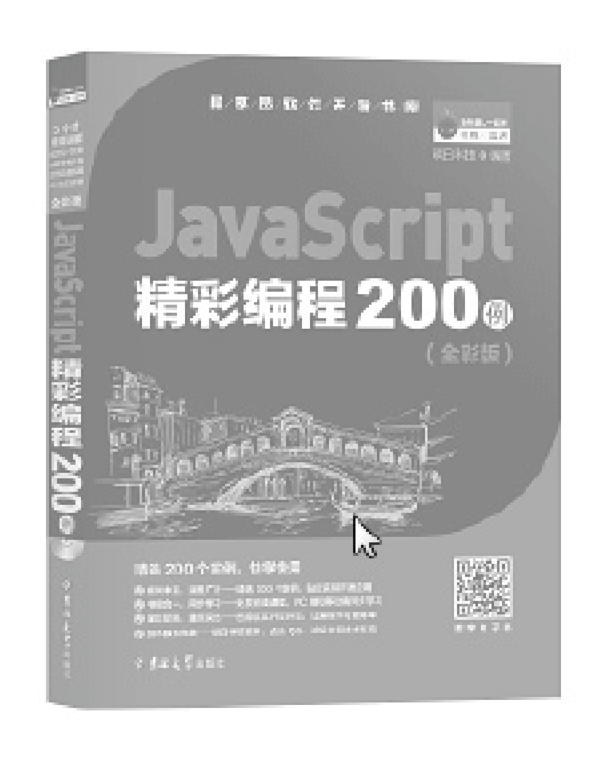


图 11.33 鼠标移入



图 11.34 鼠标移出

图 11.35 输出算式



图 11.36 计算结果

16．随意摆放的照片墙 ▷①②③④⑤⑥

实现图片在页面中随意摆放的功能。在页面中显示 3 张旋转一定角度的图片，通过拖动图片可以在页面中随意摆放照片。实现效果如图 11.37 所示。

（提示：应用鼠标事件中的 onmousedown、onmousemove 和 onmouseup 事件）



图 11.37 随意摆放的照片墙

17．模拟画图软件调整图片大小 ▷①②③④⑤⑥

应用 Windows 系统自带的画图软件，可对图片的大小进行调整。编写程序，模拟画图软件中调整图片大小的功能。效果如图 11.38 和图 11.39 所示。

（提示：应用图像对象的 width 属性和 height 属性，设置图片的宽度和高度）



图 11.38　100%显示效果



图 11.39　50%显示效果

18. 图片放大镜效果

▷①②③④⑤⑥

在电子商城中选购商品时，单击商品主图，有时会看到放大显示的商品图片。编写程序，模拟电子商城中图片放大镜的功能。当鼠标光标移到商品图片某一区域上时，右侧会对该区域进行放大显示，效果如图 11.40 所示。

（提示：应用 left 和 top 属性对元素进行定位）

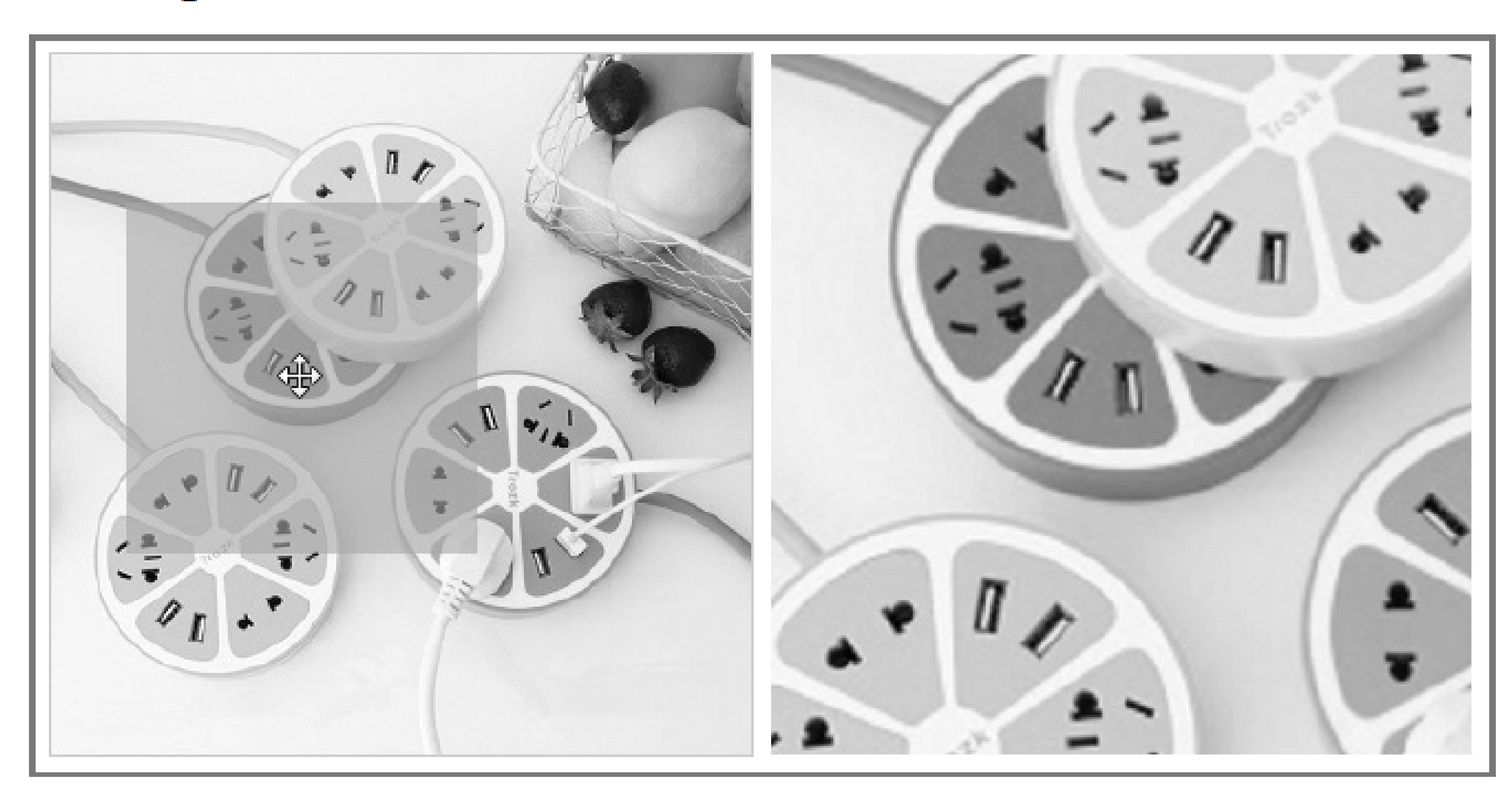


图 11.40　图片放大镜效果

19. 显示选择的酒店类型

▷①②③④⑤⑥

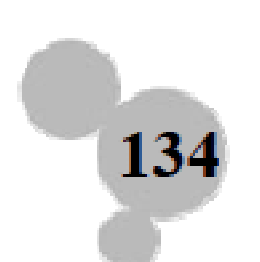
在酒店预订网站中通常会事先筛选出一些可供选择的酒店类型，如商务酒店、快捷酒店等，用户可以根据自己的需要选择适合的酒店类型。编写程序，应用复选框定义多种酒店类型，用户通过单击复选框选择需要的类型，选择的酒店类型将输出在下方页面中，效果如图 11.41 所示。

（提示：在复选框的单击事件中应用 for 语句和 checked 属性判断哪些复选框已被选中）

酒店类型：☐酒店式公寓 ☑连锁品牌 ☐家庭旅馆 ☑商务型酒店 ☐招待所 ☑客栈 ☐青年旅舍 ☐主题酒店

您选择的类型：连锁品牌 商务型酒店 客栈

图 11.41　选择酒店类型



20．切换商品类别选项卡　▷①②③④⑤⑥

设计一个可通过选项卡切换显示不同商品的网购页面。页面中有 5 个选项卡，分别代表不同的类别，当鼠标指向不同选项卡时，页面下方会显示对应的商品信息，效果如图 11.42 和图 11.43 所示。

（提示：为每个选项卡添加 onmouseover 事件，并设置当前选项卡对应的商品显示信息）



图 11.42　“推荐商品”选项卡



图 11.43　“热评商品”选项卡

21．验证用户登录信息是否为空　▷①②③④⑤⑥

设计一个简单的用户登录页面。当用户名或密码输入框为空时，在输入框右侧给出相应的提示信息；如果输入框不为空，判断用户是否登录成功。效果如图 11.44 和图 11.45 所示。

（提示：为“账号”输入框和“密码”输入框添加 onblur 事件）

登录
账 号：用户名　用户名不能为空
密 码：密码　密码不能为空
登 录

图 11.44　用户名为空

登录
登录成功！
账 号：mr
密 码：••••••
登 录

图 11.45　登录成功



22．横向导航菜单 ▷①②③④⑤⑥

在页面中输出一个横向导航菜单，默认显示“交通出行”主菜单下的子菜单。当鼠标光标指向其他主菜单时，隐藏原子菜单显示对应的子菜单，效果如图 11.46 和图 11.47 所示。

（提示：为每个主菜单添加 onmouseover 事件，并设置当前主菜单对应的子菜单）



图 11.46　默认显示效果



图 11.47　选择其他主菜单

第 12 章　绘 制 图 形



本章训练任务对应核心技术分册第 12 章“绘制图形”部分。

重点练习内容：

1. 绘制直线。
2. 绘制曲线。
3. 绘制圆形。
4. 图形平移、缩放和旋转。
5. 输出文字。

应用技能拓展学习

1. sin()方法和 cos()方法

（1）sin()方法

sin()方法是 Math 对象中的方法，可返回一个数字的正弦值。返回值范围是-1.0～1.0。语法如下：

```
Math.sin(x)
```

参数 x 为必选参数，是一个以弧度表示的角。将角度乘以 0.017453293（2PI/360），即可转换为弧度。

（2）cos()方法

cos()方法是 Math 对象中的方法，可返回一个数字的余弦值。返回值范围是-1.0～1.0。语法如下：

```
Math.cos(x)
```

参数 x 为必选参数，是一个以弧度表示的角。将角度乘以 0.017453293（2PI/360），即可转换为弧度。

例如，分别应用 sin()方法和 cos()方法输出两个弧度的正弦值和余弦值。代码如下：

```
<script type="text/javascript">
    var a=Math.sin(90*Math.PI/180);
    var b=Math.cos(Math.PI);
    document.write(a+"<br>"+b);
</script>
```

运行结果：1

−1

2. 状态的保存和恢复

状态保存和状态恢复是绘制复杂图形时必不可少的操作，需要用到 save()和 restore()方法。save()方法用于保存当前环境的状态，restore()方法用于返回之前保存过的路径状态和属性。这两个方法都没有参数。

下面来看一个示例。

```
<script type="text/javascript">
    window.onload=function(){
        var canvas = document.getElementById("canvas");
        var ctx = canvas.getContext("2d");
        ctx.fillStyle = "red";
        ctx.save();              //保存当前状态
        ctx.beginPath();
        ctx.fillRect(20,20, 100, 100);
        ctx.fillStyle = 'green';
        ctx.fillRect(150, 150, 100, 100);
        ctx.restore();           //恢复之前保存的状态
        ctx.fillRect(280, 280, 100, 100);
    }
</script>
<canvas id="canvas" width="400" height="400" style="border: 1px solid #ccc;"></canvas>
```

运行结果如图 12.1 所示。

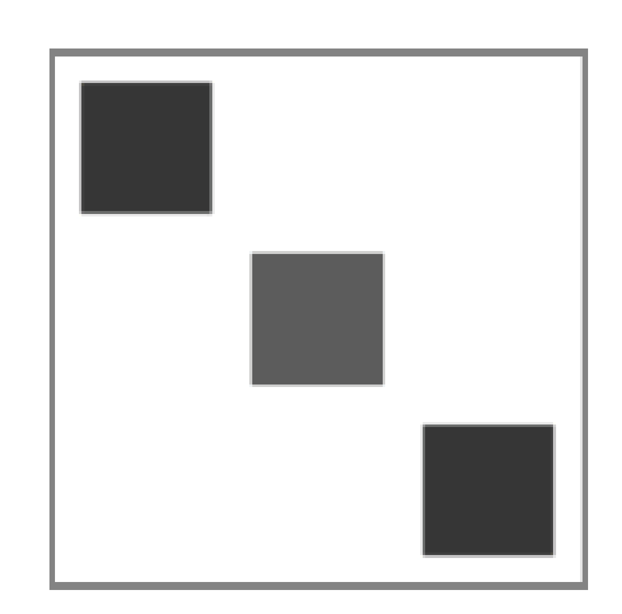


图 12.1　保存和恢复状态

3. onwheel 事件

鼠标滚轮在元素上滚动时触发 onwheel 事件。

onwheel 事件在 HTML 中的语法格式如下：

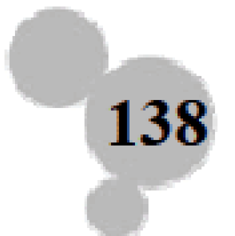
```
<element onwheel="myScript">
```

onwheel 事件在 JavaScript 中的语法格式如下：

```
object.onwheel=function(){myScript};
```

例如，当在元素上滚动鼠标滚轮时，设置元素字体大小为 36px，代码如下：

```
<div id="demo">欲穷千里目，<br>更上一层楼。</div>
```



```
<script type="text/javascript">
    document.getElementById("demo").onwheel=myFunction;
    function myFunction() {
        document.getElementById("demo").style.fontSize = "36px";
    }
</script>
```

4．zoom 属性

zoom 属性是 style 对象中的一个属性，可以设置或检索对象的缩放比例。语法如下：

```
Object.style.zoom=normal | number
```

☑ normal：使用对象的实际尺寸。

☑ number：使用百分数或者无符号浮点实数。无符号浮点实数值为 1.0 或百分数为 100%时，相当于此属性的 normal 值。

例如，在元素上滚动鼠标滚轮时，设置元素的放大比例为 300%，代码如下：

```
<div id="demo">欲穷千里目，<br>更上一层楼。</div>
<script type="text/javascript">
    document.getElementById("demo").onwheel=myFunction;
    function myFunction() {
        document.getElementById("demo").style.zoom = "300%";
    }
</script>
```

5．createPattern()方法

createPattern()方法可在指定的方向上重复指定的元素。元素可以是图片、视频或者其他<canvas>元素。语法如下：

```
context.createPattern(image,type);
```

image 参数为要重复的图片、画布或视频元素；type 参数的值必须是下面的字符串之一：

☑ repeat：默认值，表示在水平和垂直方向均重复。

☑ repeat-x：表示只在水平方向重复。

☑ repeat-y：表示只在垂直方向重复。

☑ no-repeat：表示只显示一次（不重复）。

例如，在画布中对指定图像进行重复显示。代码如下：

```
<canvas id="canvas" width="400" height="300" style="border: 1px solid #ccc;"></canvas>
<script type="text/javascript">
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext('2d');
    var image = new Image();
    image.src = "imagemr.jpg";
    image.onload = function(){
```

```
            var ptrn = context.createPattern(image,'repeat');
            context.fillStyle = ptrn;
            context.fillRect(0,0,400,300);
    };
</script>
```

运行结果如图 12.2 所示。



图 12.2　重复显示图像

6. 绘制阴影

HTML5 中，使用 canvas 元素可以给图形添加阴影效果。通过设置图形上下文对象的 4 个阴影绘制属性，可以添加不同的阴影效果。

☑ shadowOffsetX：指定阴影的横向位移量。
☑ shadowOffsetY：指定阴影的纵向位移量。
☑ shadowBlur：指定阴影边缘的模糊范围。
☑ shadowColor：指定阴影的颜色。

shadowOffsetX 和 shadowOffsetY 用来设定阴影在 X 轴和 Y 轴的延伸距离。负值表示阴影会往上或左延伸，正值则表示阴影会往下或右延伸，其默认值均为 0。

shadowBlur 用于设定阴影的模糊程度。不希望阴影的边缘太清晰时可以使用该属性。该属性值必须设定为比 0 大的数字，否则将被忽略。一般设定在 0～10 之间，开发时可以根据情况调整这个数值，以达到满意效果。

shadowColor 用于设定阴影效果的颜色，值可以是标准的 CSS 颜色值，默认是全透明的黑色。

例如，为文字“天生我材必有用”设置阴影效果，代码如下：

```
<canvas id="canvas" width="300" height="100"></canvas>
<script type="text/javascript">
    var context = document.getElementById('canvas').getContext('2d');
    context.shadowOffsetX = 3;
    context.shadowOffsetY = 3;
    context.shadowBlur = 6;
    context.shadowColor = "rgba(255, 0, 0, 0.5)";
    context.font = "30px  宋体";
    context.fillStyle = "blue";
    context.fillText("天生我材必有用", 60, 30);
</script>
```

运行结果如图 12.3 所示。

天生我材必有用

图 12.3　设置阴影效果

7．clearRect()方法

clearRect() 方法用于清空给定矩形内的指定像素。语法格式如下：

```
context.clearRect(x,y,width,height);
```

☑ x：要清除的矩形左上角的 x 坐标。
☑ y：要清除的矩形左上角的 y 坐标。
☑ width：要清除的矩形的宽度，单位为像素。
☑ height：要清除的矩形的高度，单位为像素。

例如，在给定矩形内清空一个矩形，代码如下：

```
<canvas id="canvas" width="300" height="200"></canvas>
<script type="text/javascript">
    var canvas=document.getElementById("canvas");
    var ctx=canvas.getContext("2d");
    ctx.fillStyle="green";
    ctx.fillRect(0,0,300,200);
    ctx.clearRect(50,50,100,80);
</script>
```

运行结果如图 12.4 所示。



图 12.4　清空一个矩形

8．Date 对象中的几个方法

（1）　getHours()方法

Date 对象提供的 getHours()方法可以返回时间的小时字段，返回值是 0～23 的一个整数。语法格式如下：

```
dateObj.getHours()
```

参数 dateObj 为创建的 Date 对象名。

例如，输出当前时间的小时数，代码如下：

```
<script type="text/javascript">
    var now = new Date();
    var h = now.getHours();
    document.write(h);
</script>
```

运行结果：9

（2）getMinutes()方法

Date 对象提供的 getMinutes()方法可以返回时间的分钟字段，返回值是 0～59 的一个整数。语法格式如下：

```
dateObj.getMinutes()
```

参数 dateObj 为创建的 Date 对象名。

例如，输出当前时间的分钟数，代码如下：

```
<script type="text/javascript">
    var now = new Date();
    var m = now.getMinutes();
    document.write(m);
</script>
```

运行结果：58

（3）getSeconds()方法

Date 对象提供的 getSeconds()方法可以返回时间的秒字段，返回值是 0～59 的一个整数。语法格式如下：

```
dateObj.getSeconds()
```

参数 dateObj 为创建的 Date 对象名。

例如，输出当前时间的秒数。代码如下：

```
<script type="text/javascript">
    var now = new Date();
    var s = now.getSeconds();
    document.write(s);
</script>
```

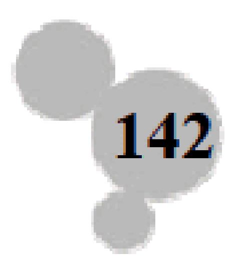
运行结果：10

（4）getMilliseconds()方法

Date 对象提供的 getMilliseconds()方法可以返回时间的毫秒字段，返回值是 0～999 的一个整数。语法格式如下：

```
dateObj.getMilliseconds()
```

参数 dateObj 为创建的 Date 对象名。



例如，输出当前时间的毫秒数，代码如下：

```
<script type="text/javascript">
    var now = new Date();
    var ms = now.getMilliseconds();
    document.write(ms);
</script>
```

运行结果：396

9．数组的 push()方法

push()方法用于向数组的末尾添加一个或多个元素，并返回添加后的数组长度。语法格式如下：

```
arrayObject.push(newelement1,newelement2,....,newelementX)
```

☑　arrayObject：必选项，表示数组名称。
☑　newelement1：必选项，表示要添加到数组的第一个元素。
☑　newelement2：可选项，表示要添加到数组的第二个元素。
☑　newelementX：可选项，表示可添加的多个元素。

例如，向数组的末尾添加一个数组元素，并输出新数组，代码如下：

```
<script type="text/javascript">
    var arr=new Array("JavaScript","HTML");
    arr.push("CSS");
    document.write(arr);
</script>
```

运行结果：JavaScript,HTML,CSS

10．在 canvas 中应用 layerX 和 layerY 属性

在 canvas 中，使用事件对象中的 layerX 和 layerY 属性可以获取元素在发生事件时，鼠标相对于 canvas 元素左上角的横坐标和纵坐标。

例如，当鼠标单击 canvas 元素时，获取鼠标在 canvas 元素中的位置。代码如下：

```
<canvas id="canvas" width="300" height="200" style="border: 1px solid #000000"></canvas>
<div id="show"></div>
<script type="text/javascript">
    var canvas=document.getElementById("canvas");
    canvas.onclick=function(ev){
        show.innerHTML="X："+ev.layerX+"，Y："+ev.layerY;
    }
</script>
```

运行结果如图 12.5 所示。

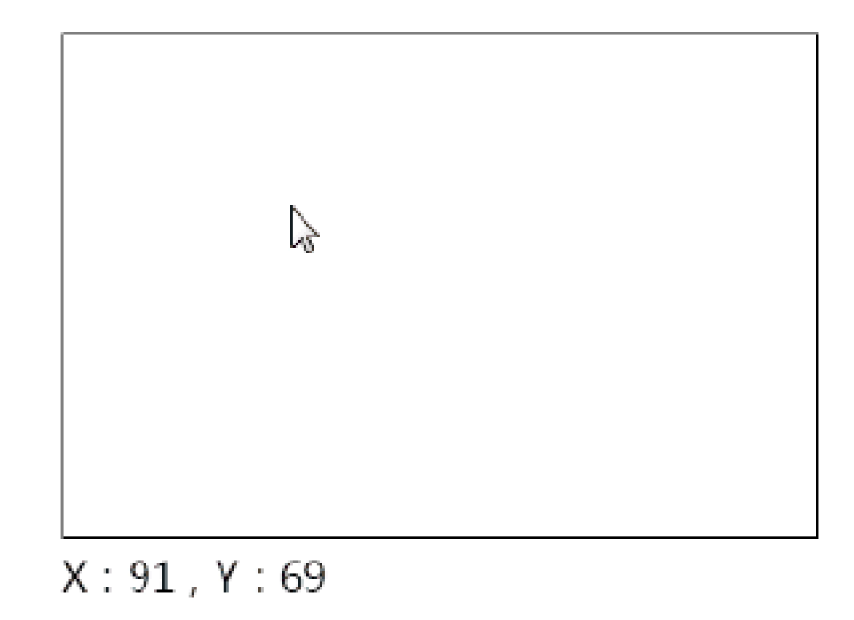


图 12.5　获取元素坐标

11．min()方法

min()方法是 Math 对象中的方法，可以返回指定数字中的最小值。语法如下：

```
Math.min(n1,n2,n3,...,nX)
```

参数 n1,n2,n3,...,nX 表示一个或多个指定数值。

例如，输出 5、6 和 9 三个数字中的最小值。代码如下：

```
<script type="text/javascript">
    document.write(Math.min(5,6,9));
</script>
```

运行结果：5

12．random()方法

Math 对象的 random 方法可返回 0（包含）～1（不包含）的一个随机数。语法如下：

```
Math.random()
```

例如，输出一个 0～1 的随机数。代码如下：

```
<script type="text/javascript">
    document.write(Math.random());
</script>
```

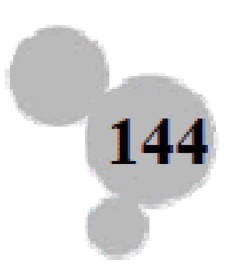
运行结果：0.979182268421426

13．requestAnimationFrame()方法

requestAnimationFrame()方法可以执行一个动画，并且要求浏览器在下次重绘之前调用指定的回调函数，以更新动画。该方法需要传入一个回调函数作为参数，该回调函数会在浏览器下一次重绘之前执行。语法格式如下：

```
window.requestAnimationFrame(callback);
```

参数 callback 为下一次重绘之前更新动画帧所调用的函数。



例如，应用 requestAnimationFrame()方法使数字不断地快速执行加 1 操作，代码如下：

```
<div id="demo"></div>
<script type="text/javascript">
    var oDiv = document.getElementById('demo');
    var num = 0;
    function account(){
        num++;
        oDiv.innerHTML = num;
        requestAnimationFrame(account);
    }
    account();
</script>
```

实战技能强化训练

训练一：基本功强化训练

1. 绘制火柴人　▷①②③④⑤⑥

绘制火柴人时，将人抽象成火柴的形状，头部类似于火柴头，而身体和四肢类似于火柴杆。编写程序，应用 canvas 绘制一个火柴人，实现效果如图 12.6 所示。

2. 绘制红心　▷①②③④⑤⑥

贝塞尔曲线常用于绘制复杂而有规律的形状。编写程序，使用三次方贝塞尔曲线绘制一个红色实心的红心。实现效果如图 12.7 所示。

3. 绘制简易房屋　▷①②③④⑤⑥

应用 canvas 绘制一个简易的房屋。该房屋包括屋顶、房子主体、门和窗等几部分，效果如图 12.8 所示。

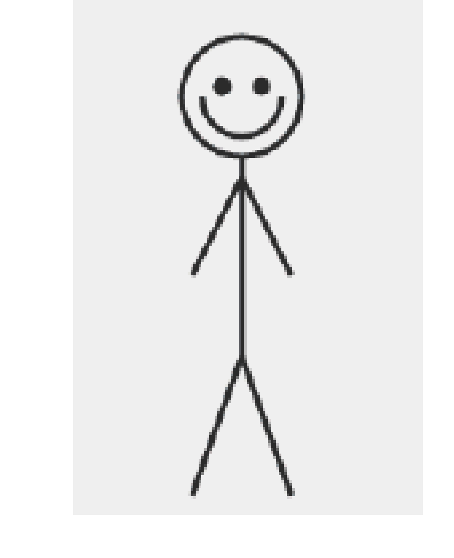


图 12.6　绘制火柴人



图 12.7　绘制红心

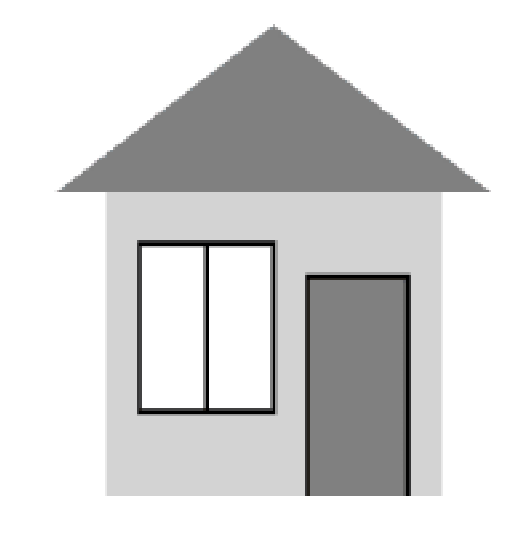


图 12.8　绘制简易房屋

4. 在图片上输出文字 ▷①②③④⑤⑥

在图片上显示填充文字。将文字“一幅美丽的风景画”输出在图片中，实现效果如图 12.9 所示。

5. 绘制向日葵 ▷①②③④⑤⑥

应用 canvas 绘制一个向日葵，效果如图 12.10 所示。

（提示：在 for 语句中应用 lineTo()方法绘制直线）

6. 在画布中显示图片 ▷①②③④⑤⑥

通过 canvas 还可以读取磁盘或网络中的图像文件，然后使用 canvas API 将图像绘制在画布中。编写程序，在 canvas 画布中显示一张指定的图片，效果如图 12.11 所示。



图 12.9 在图片上输出文字

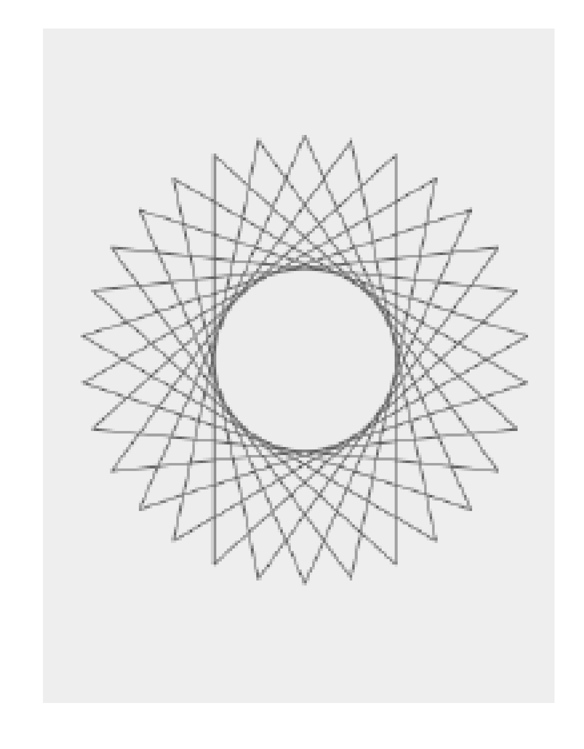


图 12.10 绘制向日葵



图 12.11 在画布中显示图片

7. 通过鼠标滚轮放大和缩小图片 ▷①②③④⑤⑥

在 canvas 画布中显示一张指定的图片，并使用鼠标滚轮放大、缩小图片。当用户将鼠标指针移动到图片上时，图片会获取焦点，滚动鼠标滚轮，即可改变图片的大小，实现效果如图 12.12 和图 12.13 所示。

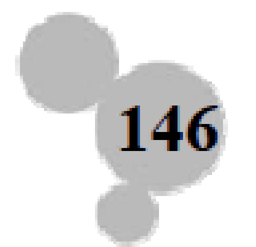
（提示：应用 onwheel 事件、zoom 属性和 scale()方法）



图 12.12 放大图片



图 12.13 缩小图片



8. 制作图像平铺效果　▷①②③④⑤⑥

所谓图像平铺，就是使用按一定比例缩小后的图像将画布填满。编写程序，应用 canvas 实现图像的平铺效果，如图 12.14 所示。

9. 制作文字阴影效果　▷①②③④⑤⑥

为了使文字更加美观，可为文字添加阴影效果。编写程序，为文字添加阴影效果，如图 12.15 所示。

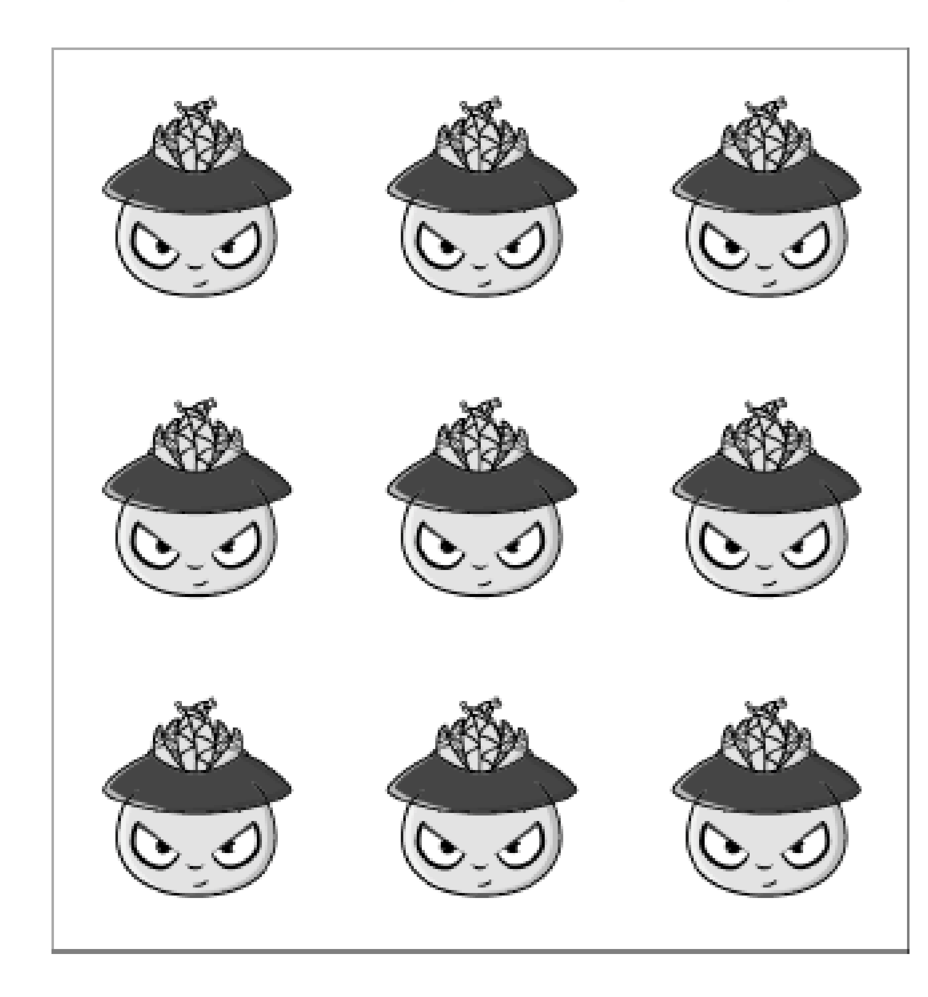


图 12.14　图像平铺效果



图 12.15　文字阴影效果

训练二：实战能力强化训练

10. 绘制别墅与树木　▷①②③④⑤⑥

应用 canvas 绘制一幢别墅，在别墅两旁绘制两棵树，实现效果如图 12.16 所示。

11. 制作桌面时钟　▷①②③④⑤⑥

试着应用图形路径、变形以及动画等知识制作一个动态的桌面时钟，实现效果如图 12.17 所示。

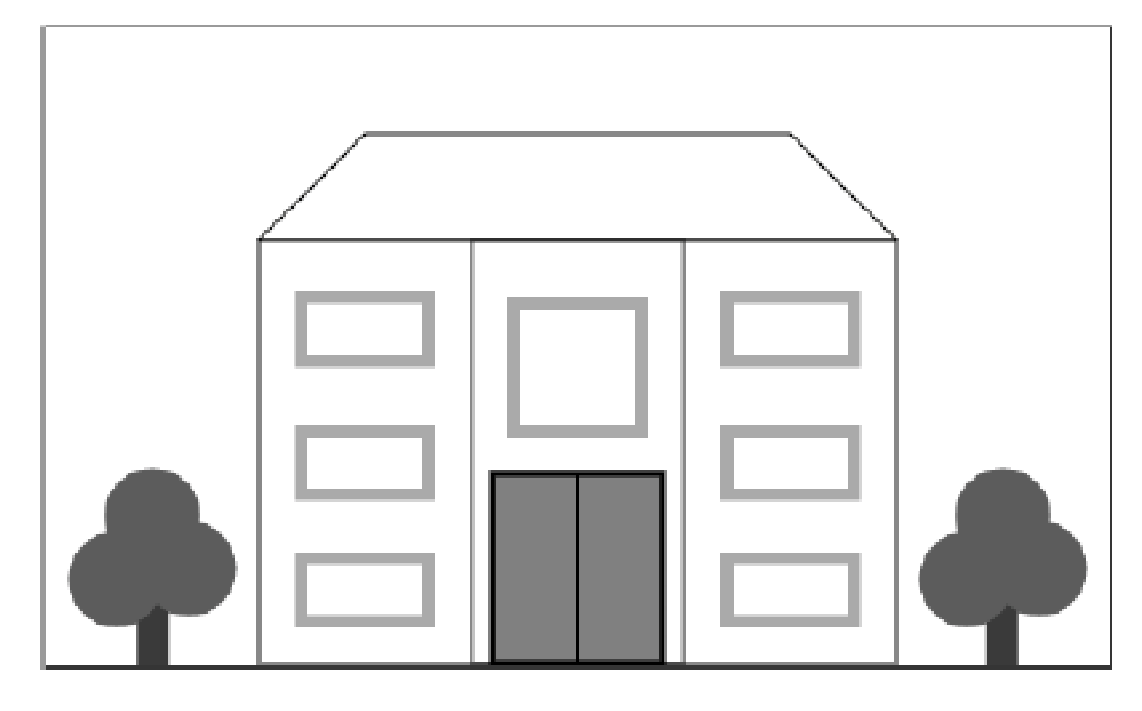


图 12.16　绘制别墅与树木



图 12.17　桌面时钟

12. 小车的移动和停止 ▷①②③④⑤⑥

编写程序，实现小车前进和停止的动画效果。当单击“前进”按钮时，小车开始向前移动；当单击“停止”按钮时，小车停止移动。实现效果如图12.18和图12.19所示。

（提示：应用setInterval()方法和translate()方法）



图12.18 单击开始前进



图12.19 停止前进

13. 图像放大镜 ▷①②③④⑤⑥

实现一个图像放大镜的效果。当鼠标指向图片时，鼠标右下角将对当前区域进行放大显示，效果如图12.20和图12.21所示。

（提示：应用onmousemove 事件和drawImage()方法）



图12.20 原图效果



图12.21 局部放大效果

14．制作简易写字板　　▷①②③④⑤⑥

编写程序，制作一个简易的写字板，写字效果如图 12.22 所示。

（提示：为 canvas 画布分别应用 onmousedown、onmousemove 和 onmouseup 事件）



图 12.22　写作板效果

15．图片渐隐渐现效果　　▷①②③④⑤⑥

在 canvas 画布中显示一张指定的图片，使图片从模糊状态变为清晰状态，然后又从清晰状态变为模糊状态。重复执行此操作，使图片形成渐隐渐现的效果，如图 12.23 和图 12.24 所示。

（提示：应用 setInterval()方法和 style 对象中的 opacity 属性）



图 12.23　模糊图片



图 12.24　清晰图片

16．图像局部放大效果　　▷①②③④⑤⑥

编写程序，实现图像局部放大的功能。在页面左侧的图像中拖动鼠标选择准备放大的区域，在页面右侧会显示该区域放大后的图像，效果如图 12.25 所示。

（提示：为 canvas 画布应用 onmousedown、onmousemove 和 onmouseup 事件）



图 12.25　图像局部放大效果

17．下雪动画效果

▷①②③④⑤⑥

编写程序，模拟下雪的动画效果，如图 12.26 所示。

（提示：在 for 语句中应用 arc()方法绘制不同半径的圆形）



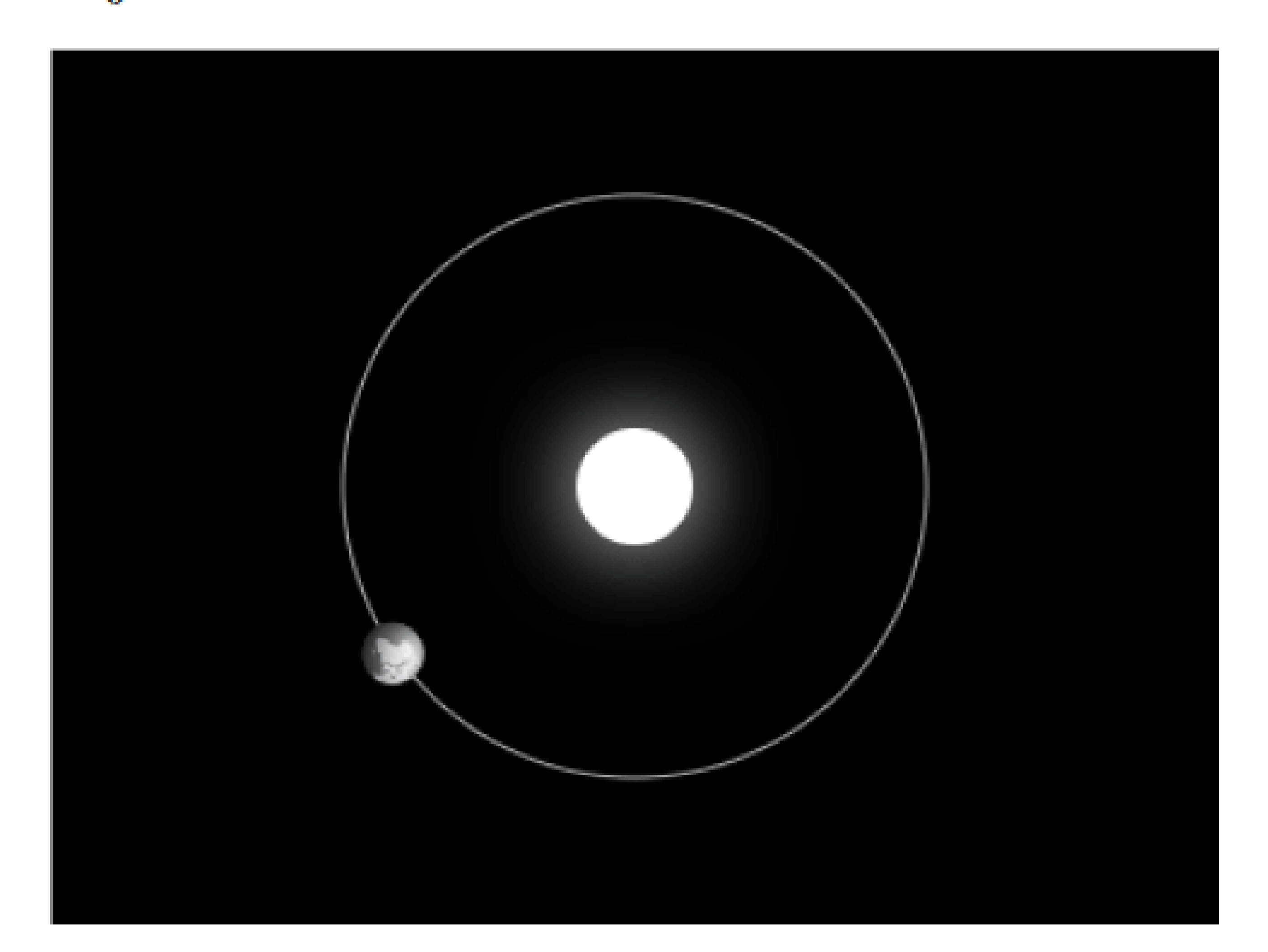
图 12.26　下雪动画效果

18．地球绕太阳公转效果

▷①②③④⑤⑥

编写程序，模拟地球围绕太阳公转的效果。页面中，太阳的位置不变，地球按照一定的速度沿着圆形轨道不断旋转，实现效果如图 12.27 所示。

（提示：应用 rotate()方法旋转坐标系，应用 drawImage()方法绘制地球图像，应用 requestAnimation Frame()方法实现动画效果）



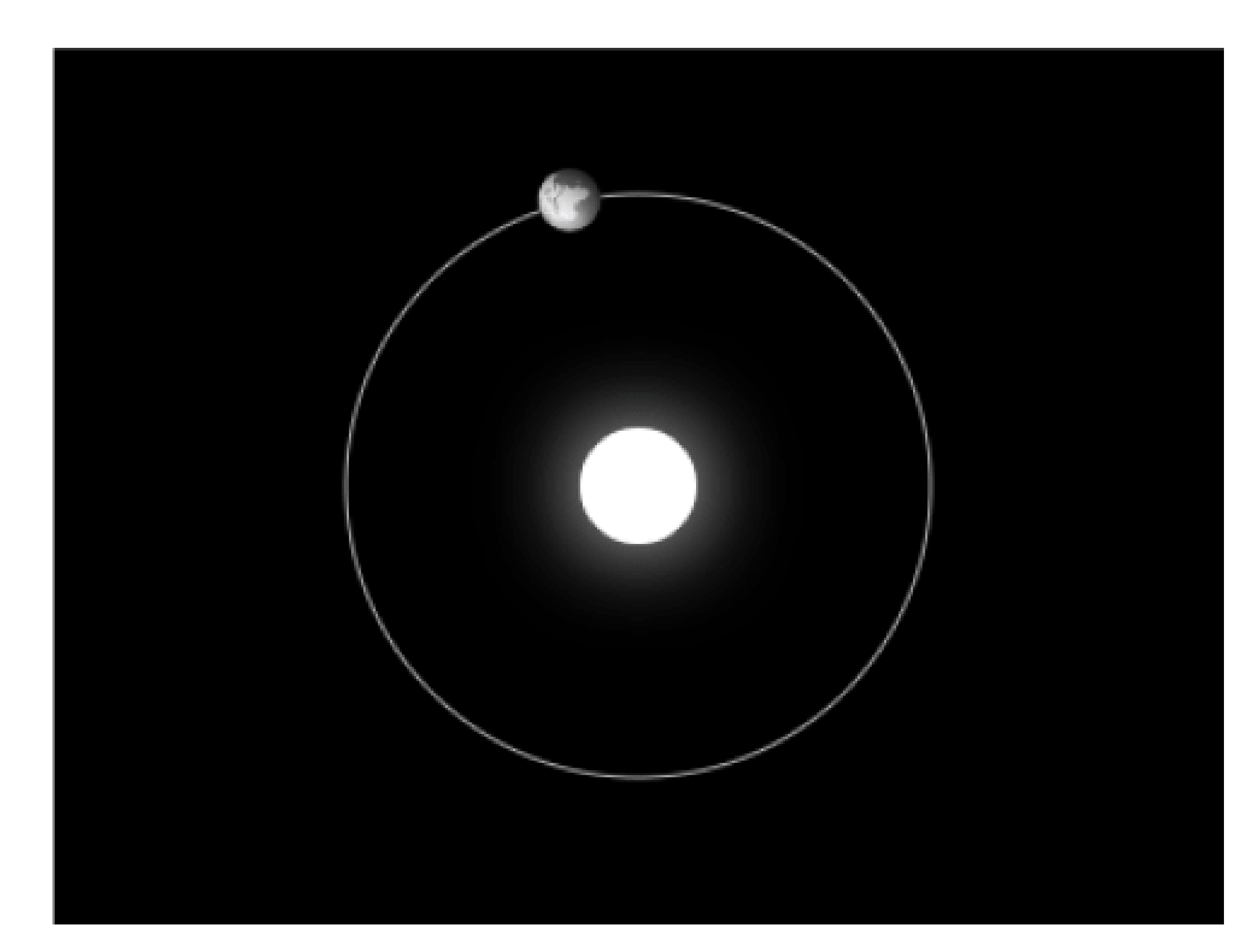


图 12.27　模拟公转效果

第 13 章　文件与拖放


学习指南

本章训练任务对应核心技术分册第 13 章“文件与拖放”部分。

重点练习内容：

1. 应用file对象获取文件。
2. 获取文件的类型与大小。
3. 预览图片。
4. 读取文本文件。
5. 拖放文件。

应用技能拓展学习

1. Math 对象的两个方法

（1）log()方法

log()方法用于返回一个数的自然对数（基于 E）。语法格式如下：

```
Math.log(x)
```

参数 x 为必选项，表示一个任意数值或表达式。

例如，获取 6 的自然对数，代码如下：

```
<script type="text/javascript">
    document.write(Math.log(6));
</script>
```

运行结果：1.791759469228055

（2）pow()方法

pow()方法用于返回 x 的 y 次幂的值。语法格式如下：

```
Math.pow(x,y)
```

☑ x：必选项，表示底数，必须是数字。

☑ y：必选项，表示幂数，必须是数字。

例如，获取 2 的 6 次幂的值。代码如下：

```
<script type="text/javascript">
    document.write(Math.pow(2,6));
```

```
</script>
```

运行结果：64

2. 正则表达式简介

正则表达式描述了一种字符串匹配模式，即用户可使用一系列由普通字符和特殊字符构建的能够明确描述文本字符串的匹配模式，检查一个字符串是否含有某种子字符串，将匹配的子字符串替换掉，或者从某个字符串中取出符合一定条件的子字符串等。

正则表达式是一种可以用于模式匹配和替换的强有力的工具。其作用如下：

☑ 测试字符串的某个模式。例如，可以对输入的字符串进行测试，测试该字符串是否存在一个电话号码模式或一个信用卡号码模式。这称为数据有效性验证。

☑ 替换文本。可以在文档中使用一个正则表达式来标识特定文字，然后将其全部删除，或者替换为别的文字。

☑ 根据模式匹配从字符串中提取一个子字符串。例如，在文本或输入字段中查找特定的文字。

在 JavaScript 中，正则表达式是由 RegExp 对象表示的，利用 RegExp 对象可完成有关正则表达式的所有操作和功能。

（1）创建 RegExp 对象

每一条正则表达式模式对应一个 RegExp 实例。有两种方式可以创建 RegExp 对象的实例：一种是使用 RegExp 构造函数创建，另一种是采用纯文本格式创建。

使用 RegExp 构造函数创建 RegExp 对象的语法如下：

```
new RegExp("pattern"[,"flags"])
```

使用纯文本格式创建 RegExp 对象的语法如下：

```
/pattern/[flags]
```

上述两种语法中，pattern 部分为要使用的正则表达式模式文本，是必需的。在第一种方式中，pattern 部分以 JavaScript 字符串的形式存在，需要使用双引号或单引号括起来；在第二种方式中，pattern 部分嵌套在两个“/”之间，不能使用引号。

flags 部分设置正则表达式的标志信息，是可选项。如果设置 flags 部分，在第一种方式中，以字符串的形式存在；在第二种方式中，以文本的形式紧接在最后一个“/”字符之后。flags 可以是以下标志字符的组合。

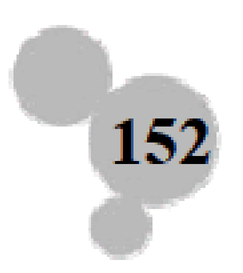
☑ g：全局标志。如果设置了这个标志，对某个文本执行搜索和替换操作时，将对文本中所有匹配的部分起作用。如果不设置这个标志，则仅搜索和替换最早匹配的内容。

☑ i：忽略大小写标志。如果设置了这个标志，进行匹配比较时，将忽略大小写。

（2）RegExp 对象的 test()方法

该方法用于检查被查找字符串中是否存在指定的模式。如果存在，则返回 true，否则就返回 false。语法如下：

```
rgexp.test(str)
```



☑ rgexp：必选项，包含正则表达式模式或可用标志的正则表达式对象。

☑ str：必选项，表示被查找的字符串。

例如，判断字符串“HTML+CSS+JavaScript”中是否包含子字符串“JavaScript”，代码如下：

```
<script type="text/javascript">
    var myString="HTML+CSS+JavaScript";
    var myregex = new RegExp("JavaScript");
    if (myregex.test(myString)){
        document.write("找到了匹配的模式");
    }else{
        document.write("未找到匹配的模式");
    }
</script>
```

运行结果：找到了匹配的模式

3. naturalWidth 和 naturalHeight 属性

在 HTML5 中新增了 naturalWidth 和 naturalHeight 两个属性。其中，naturalWidth 属性可以获取图片的原始宽度，naturalHeight 属性可以获取图片的原始高度。

例如，获取指定图片的原始宽度和高度，代码如下：

```
<img id="myimg" src="face.png" width="200" height="200">
<script type="text/javascript">
    window.onload = function () {
        var Oimg = document.getElementById("myimg");
        document.write("原始宽度："+Oimg.naturalWidth+"，原始高度："+Oimg.naturalHeight);
    }
</script>
```

运行结果：原始宽度：219，原始高度：231

4. substr()方法

String 对象的 substr()方法可从字符串的指定位置开始，提取指定长度的子字符串。语法如下：

```
stringObject.substr(start,length)
```

☑ stringObject：String 对象名或字符变量名。

☑ start：必选参数。指定要提取的字符串片断的开始位置。该参数可以是负数，如果是负数，则从字符串的尾部开始算起。

☑ length：可选参数。用于指定提取的子字符串的长度。如果省略该参数，表示结束位置为字符串的最后一个字符。

注意：由于浏览器的兼容性问题，因此 substr()方法的第一个参数不建议使用负数。

例如，在字符串“你好 JavaScript”中，从下标为 2 的字符开始提取到字符串末尾。代码如下：

```
<script type="text/javascript">
    var str="你好 JavaScript";                    //定义字符串
    document.write(str.substr(2));
</script>
```

运行结果：JavaScript

例如，在字符串“你好 JavaScript”中，从下标为 2 的字符开始提取 4 个字符。代码如下：

```
<script type="text/javascript">
    var str="你好 JavaScript";                    //定义字符串
    document.write(str.substr(2,4));
</script>
```

运行结果：Java

5. canvas 元素的 toBlob()方法

toBlob()方法可以创造 Blob 对象，用于展示 canvas 上的图片，且该图片文件可以被缓存或保存到本地。如不特别指明，图片的格式默认为 image 或 png，分辨率为 96dpi。语法如下：

```
canvas.toBlob(callback, type, encoderOptions);
```

☑ callback：回调函数，可获得一个单独的 Blob 对象参数。
☑ type：String 类型，用于指定图片格式，默认格式为 image 或 png。
☑ encoderOptions：Number 类型，取值范围为 0～1。当请求图片格式为 image 或 jpeg 时，用来指定图片的输出质量。如果这个参数的值不在指定类型与范围之内，则使用默认值，其余参数将被忽略。

6. URL.createObjectURL()方法

URL.createObjectURL()方法会根据传入的参数创建一个指向该参数对象的 URL。这个 URL 的生命周期仅存在于它被创建的这个文档中。这个新的 URL 对象表示指定的 File 对象或者 Blob 对象。语法如下：

```
objectURL = URL.createObjectURL(object);
```

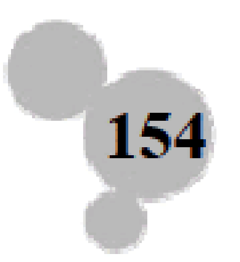
参数 object 为用于创建 URL 的 File 对象或 Blob 对象。

7. URL.revokeObjectURL()方法

URL.revokeObjectURL()方法用来释放一个之前已经存在的、通过调用 URL.Create ObjectURL()方法创建的 URL 对象。语法如下：

```
window.URL.revokeObjectURL(objectURL);
```

参数 objectURL 表示通过调用 URL.createObjectURL()方法产生的 URL 对象。



例如，通过 canvas 图像复制一张 png 格式的图片，并显示在页面中。代码如下：

```
<canvas id="canvas" width="200" height="200"></canvas>
<script type="text/javascript">
    var canvas = document.getElementById("canvas");
    var context = canvas.getContext('2d');
    context.fillStyle = 'lightgreen';
    context.fillRect(50, 50,100, 100);
    canvas.toBlob(function(blob) {
        var newImg = document.createElement("img");
        var url = URL.createObjectURL(blob);
        newImg.onload = function() {
            URL.revokeObjectURL(url);
        };
        newImg.src = url;
        document.body.appendChild(newImg);
    });
</script>
```

运行结果如图 13.1 所示。

8. createElement()方法

createElement()方法可根据指定的名称来创建一个 HTML 元素。语法如下：

```
sElement=document.createElement(sName)
```

☑ sElement：用来接收该方法返回的一个对象。
☑ sName：创建的元素名称。

9. appendChild()方法

appendChild()方法可向节点的子节点列表末尾添加一个新的子节点。语法如下：

```
obj.appendChild(node)
```

参数 node 为必选参数，表示要添加的节点对象。
例如，将创建的节点添加到列表中，代码如下：

```
<ul id="myList">
    <li>HTML</li>
    <li>CSS</li>
</ul>
<button onclick="myFunction()">添加列表项</button>
<script type="text/javascript">
    function myFunction(){
        var node=document.createElement("li");
        var textnode=document.createTextNode("JavaScript");
        node.appendChild(textnode);
```

```
            document.getElementById("myList").appendChild(node);
    }
</script>
```

运行程序，添加节点前的效果如图 13.2 所示，添加节点后的效果如图 13.3 所示。



图 13.1　复制图片效果　　图 13.2　添加节点前　　图 13.3　添加节点后

实战技能强化训练

训练一：基本功强化训练

1．显示上传文件名　　▷①②③④⑤⑥

使用 File 对象输出上传文件名。单击“选择文件”按钮，在弹出的对话框中选择要上传的文件；选择文件后单击“打开”按钮，关闭对话框；然后单击“显示文件名称”按钮，将弹出一个对话框，在这个对话框中显示上传文件的名称，实现效果如图 13.4 和图 13.5 所示。



图 13.4　选择上传文件

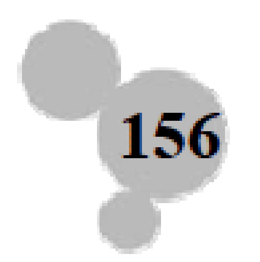




图 13.5　显示上传文件信息

2. 单图缩略图预览　▷①②③④⑤⑥

实现预览指定大小图片的功能。单击“选择文件”按钮，在弹出的对话框中选择要上传的图片文件，如图 13.6 所示，然后单击“打开”按钮并关闭对话框，此时在页面中会显示指定大小的图片，如图 13.7 所示。

（提示：应用 readAsDataURL()方法预览图片文件）

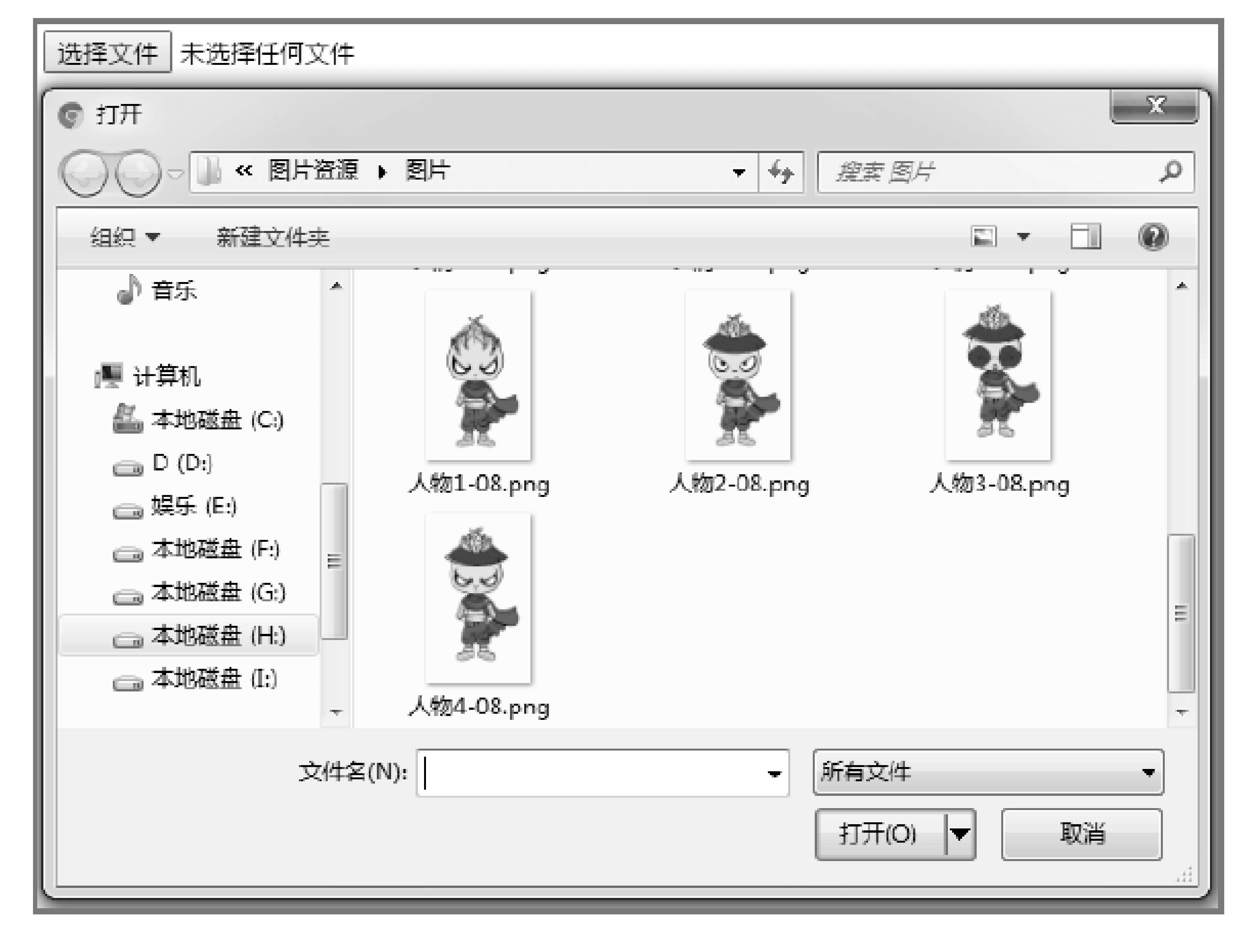


图 13.6　选择要上传的图片



图 13.7　显示缩略图

3. 单图预览并输出文件信息　▷①②③④⑤⑥

对单个上传图片进行预览，并获取上传图片信息。单击“选择文件”按钮，在弹出的对话框中选择要上传的图片文件，然后单击“打开”按钮并关闭对话框，页面中将输出图片以及图片的大小、类型、尺寸等信息，实现效果如图 13.8 和图 13.9 所示。

（提示：应用 readAsDataURL()方法预览图片文件）

图 13.8　选择上传图片

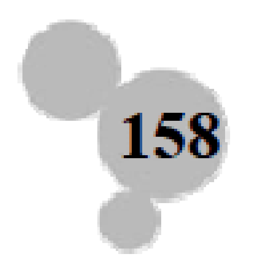


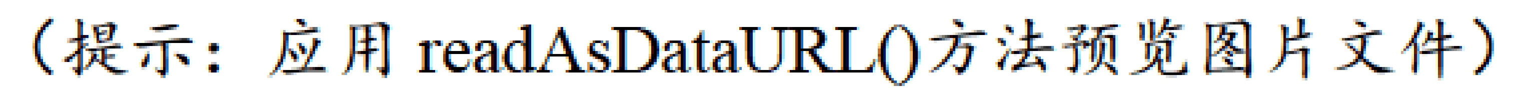
图 13.9　获取图片信息

4．编辑照片墙

▷①②③④⑤⑥

在有些博客类网站中提供了照片墙功能，其中可以显示用户已经上传的照片。编写程序，实现预览用户上传照片的功能。单击“选择文件”按钮，选择要上传的照片。然后单击“打开”按钮关闭对话框，单击“我选好了”按钮，用户选择的照片将显示在照片墙中。单击某张照片右上角的关闭按钮，还可以将其删除。实现效果如图 13.10 和图 13.11 所示。





（提示：应用 readAsDataURL()方法预览图片文件）

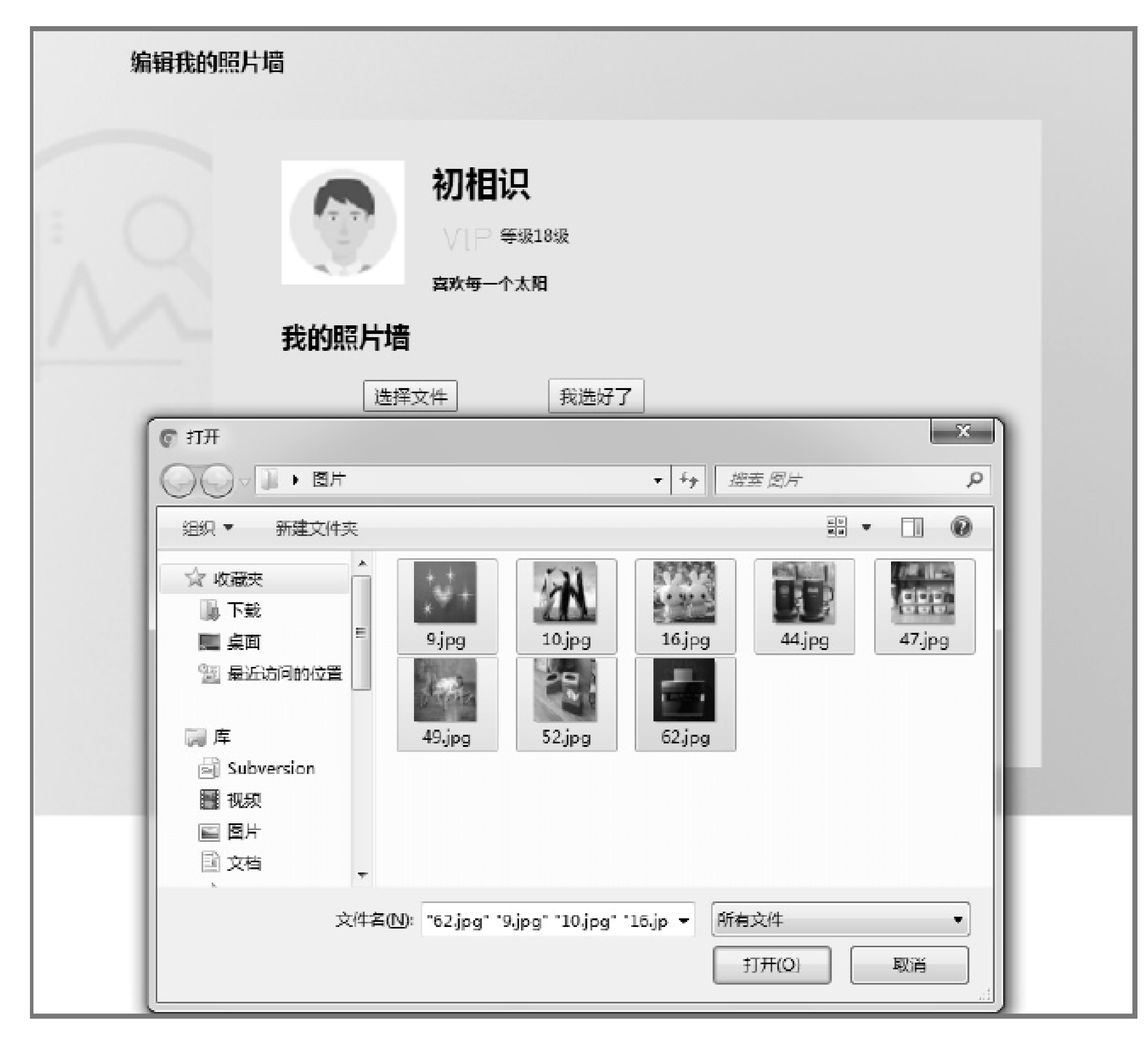


图 13.10　选择照片



图 13.11　照片墙效果

5. 文本的上传预览　▷①②③④⑤⑥

实现文本的上传预览功能。单击“选择文件”按钮，选择要浏览的文本文件，然后单击“打开”按钮关闭对话框。单击“查看文件”按钮，弹出确认对话框，单击“确定”按钮，文本文件的内容将在页面中显示出来。实现效果如图 13.12 和图 13.13 所示。

（提示：应用 readAsDataURL()方法预览图片文件，应用 readAsText()方法读取文本文件）



图 13.12　查看文件

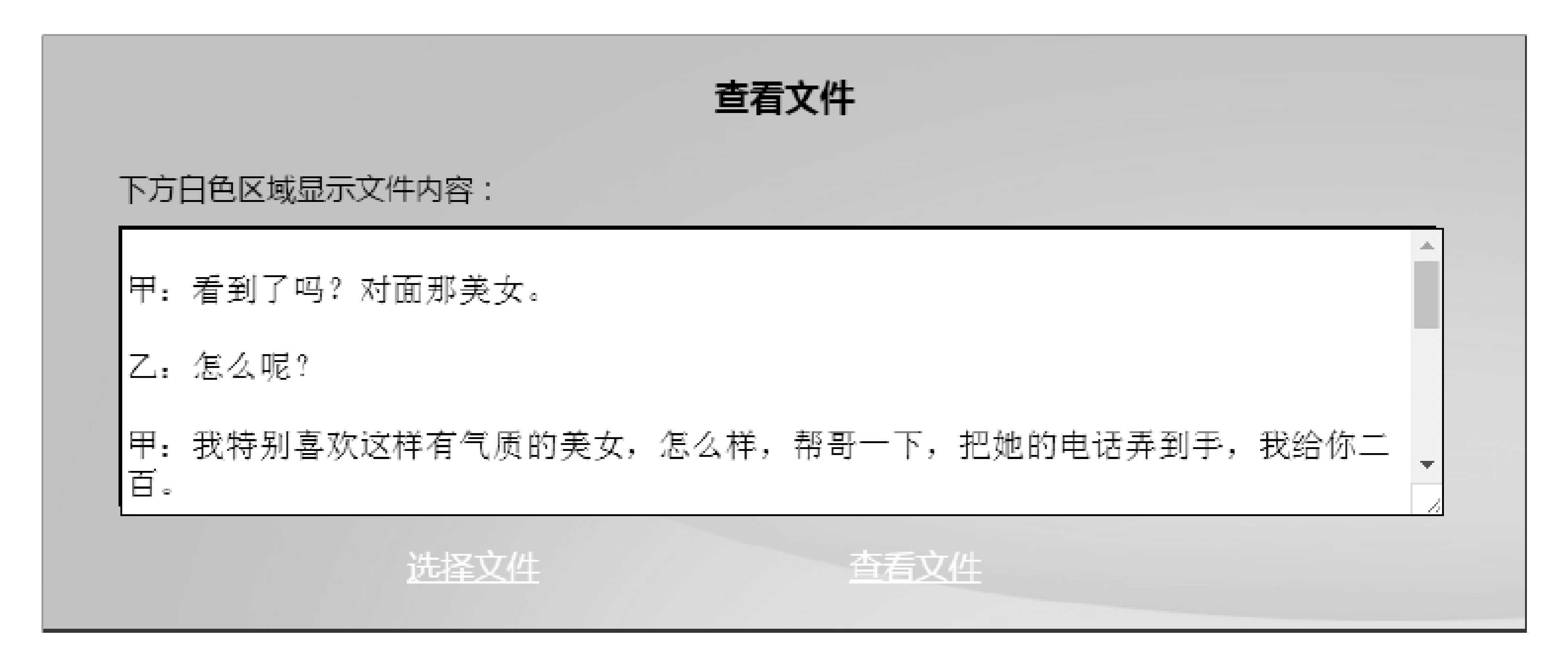


图 13.13　文件上传后效果

6. 应用事件属性拖放图片到指定区域　▷①②③④⑤⑥

应用事件属性，将页面中的元素拖放到指定区域。页面中有一个矩形区域，矩形区域的右侧有一张图片。按住鼠标左键可以拖动图片，并且该图片只能被拖放到左侧的矩形区域中，实现效果如图 13.14 和图 13.15 所示。

（提示：在 ondrop 事件的处理函数中应用 appendChild()方法）

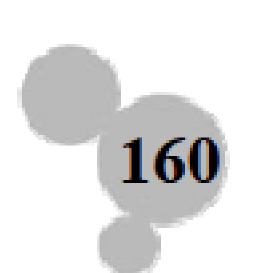




图 13.14　拖动图片前



图 13.15　拖动图片后

7. 应用事件监听器帮小鸟回家　▷①②③④⑤⑥

编写程序，模拟帮小鸟回家的游戏效果。拖动左上角的小鸟图片时，出现“小鸟正在回家…”文字。将其拖动到右下方房屋中时，出现“小鸟已经回家！”文字，效果如图 13.16 和图 13.17 所示。



图 13.16　初始效果



图 13.17　小鸟已回家

8. 将文件拖动到指定区域预览　▷①②③④⑤⑥

编写程序，实现文件拖拽到指定区域后自动预览文件的功能。在页面中显示一个文件预览区域，当把其他窗口中的图片文件或 TXT 文本文件拖拽到该区域时，会自动打开该文件进行预览。实现效果如图 13.18 和图 13.19 所示。

（提示：为目标元素添加 dragover 事件和 drop 事件）



图 13.18　选择文件



图 13.19　将文件拖动到指定区域

训练二：实战能力强化训练

9. 将商品拖至购物车　▷①②③④⑤⑥

编写程序，实现将图书商品拖入购物车的效果。在页面中输出若干图书商品的图片，应用鼠标将图书直接拖放到购物车中，此时图书的书名、定价、数量和总价等信息将显示在下面的列表中，效果如图 13.20 和图 13.21 所示。

（提示：在 dragstart 事件中应用 setData()方法为元素添加数据，在 drop 事件中应用 getData()方法获取数据）



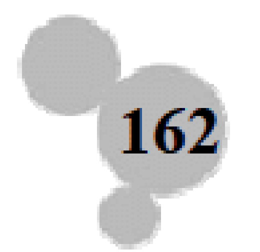
图 13.20　购物车为空



图 13.21　将图书拖至购物车

10. 随意拖动广告图　▷①②③④⑤⑥

编写程序，实现页面中广告可随意拖动的效果。在页面左上角显示一张广告图片，按住鼠标左键，



可将广告图片拖动到页面中的任何位置，如图 13.22 和图 13.23 所示。

（提示：应用鼠标事件中的 onmousedown、onmousemove 和 onmouseup 事件）







图 13.22　随意拖动广告（1）　　图 13.23　随意拖动广告（2）

11. 下载 canvas 图像　▷①②③④⑤⑥

编写程序，实现下载 canvas 图像的功能。在页面中输出一张 canvas 图像，单击“下载”按钮，该图像会被下载到本地计算机中，效果如图 13.24 所示。

（提示：设置<a>标签的 download 属性）

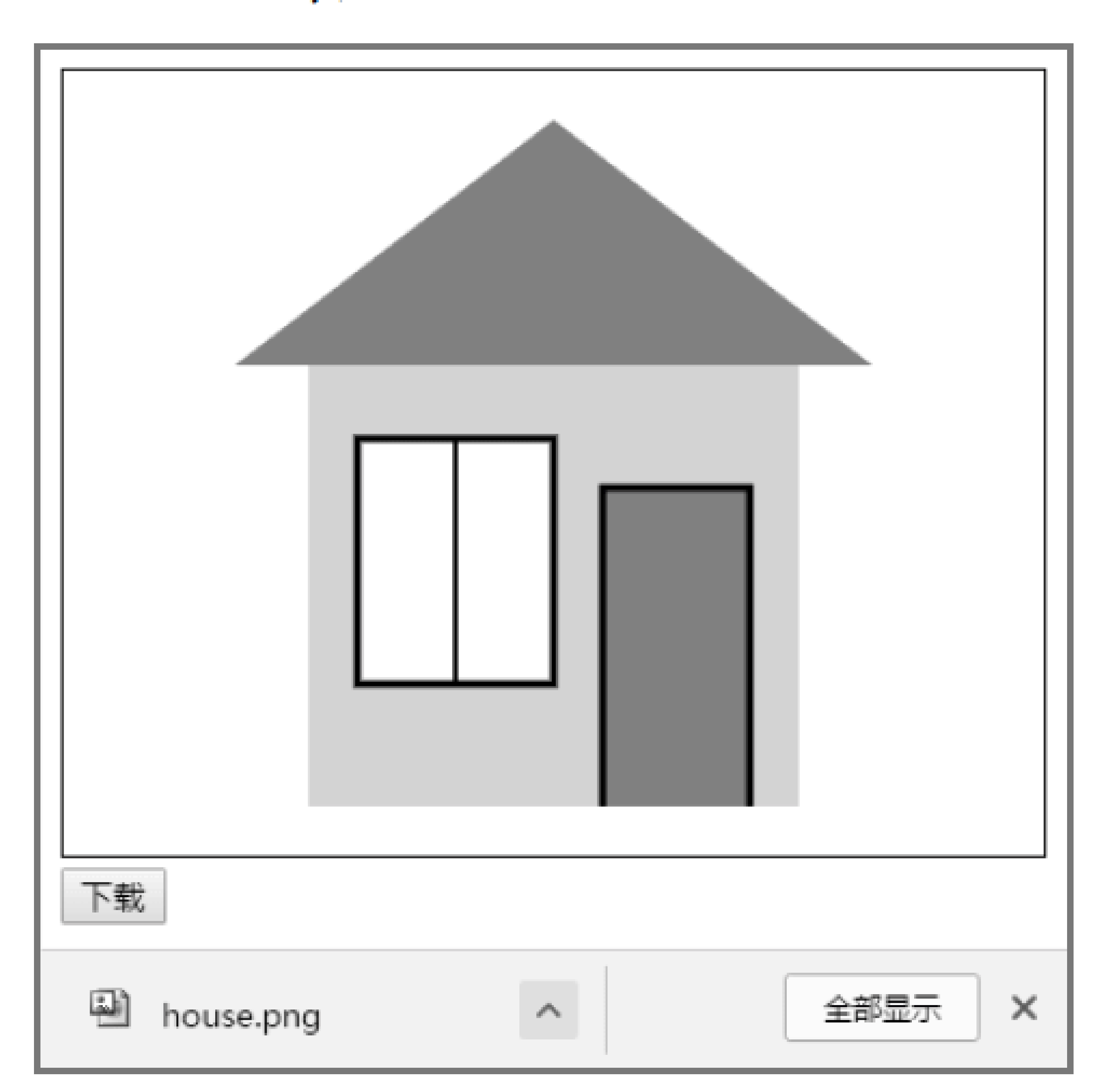


图 13.24　下载 canvas 图像

12. 预览多个上传文件信息　▷①②③④⑤⑥

编写程序，对多个上传图片进行预览，并获取上传图片的名称、大小信息。单击“选择文件”按钮，选择要上传的多个图片文件，然后单击“打开”按钮并关闭对话框后，会在页面中输出图片的缩略图以及图片的名称、大小等信息，如图 13.25 和图 13.26 所示。



图 13.25　选择多个上传图片

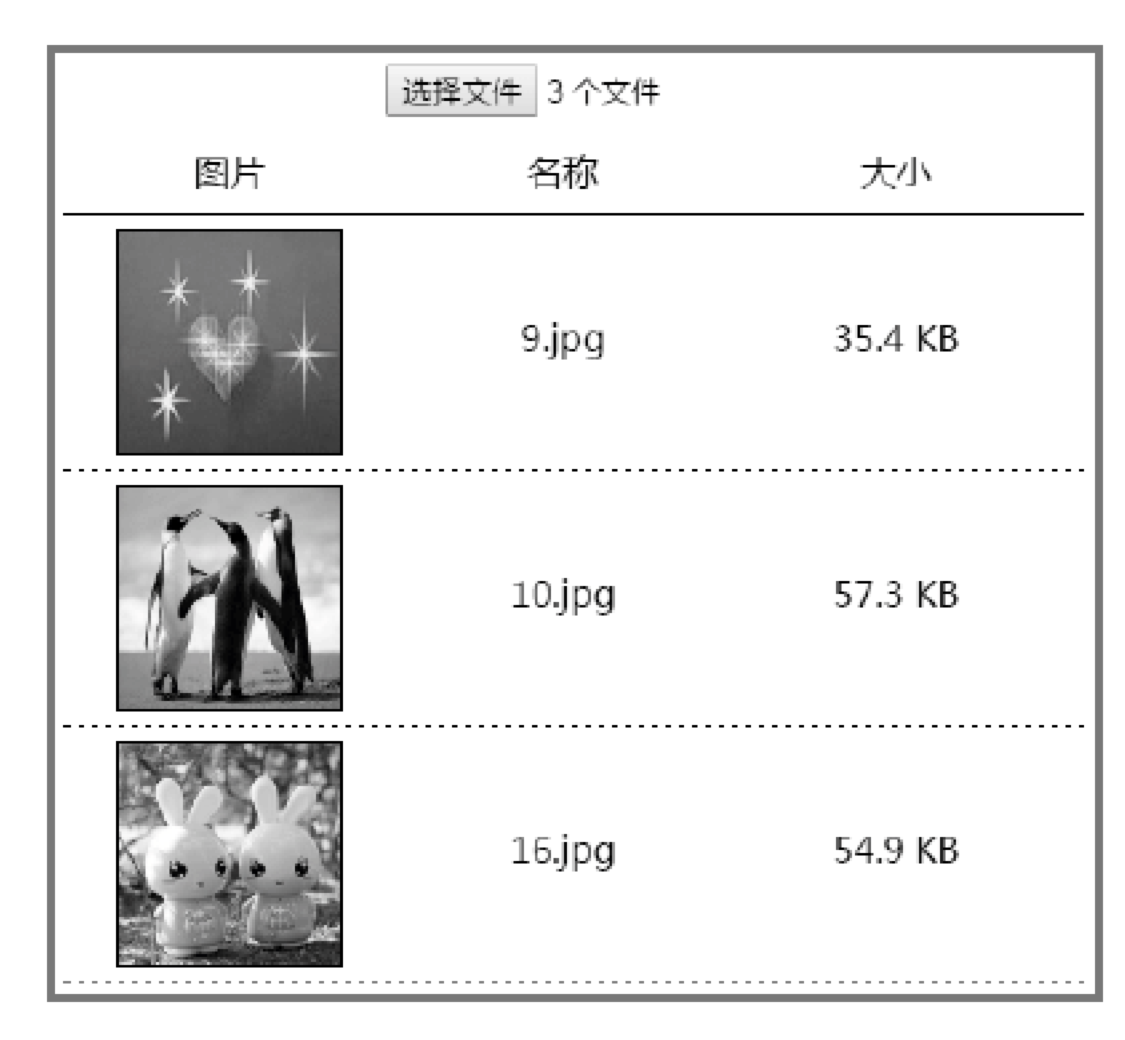


图 13.26　预览多个上传图片

13. 图片在两个容器内随意拖动　　▷①②③④⑤⑥

编写程序，实现图片在两个容器内的随意拖动。在页面左侧有一张图片，应用鼠标可以将该图片拖动到右侧任意容器当中，如图 13.27 和图 13.28 所示。

（提示：在 ondrop 事件的处理函数中应用 appendChild()方法）

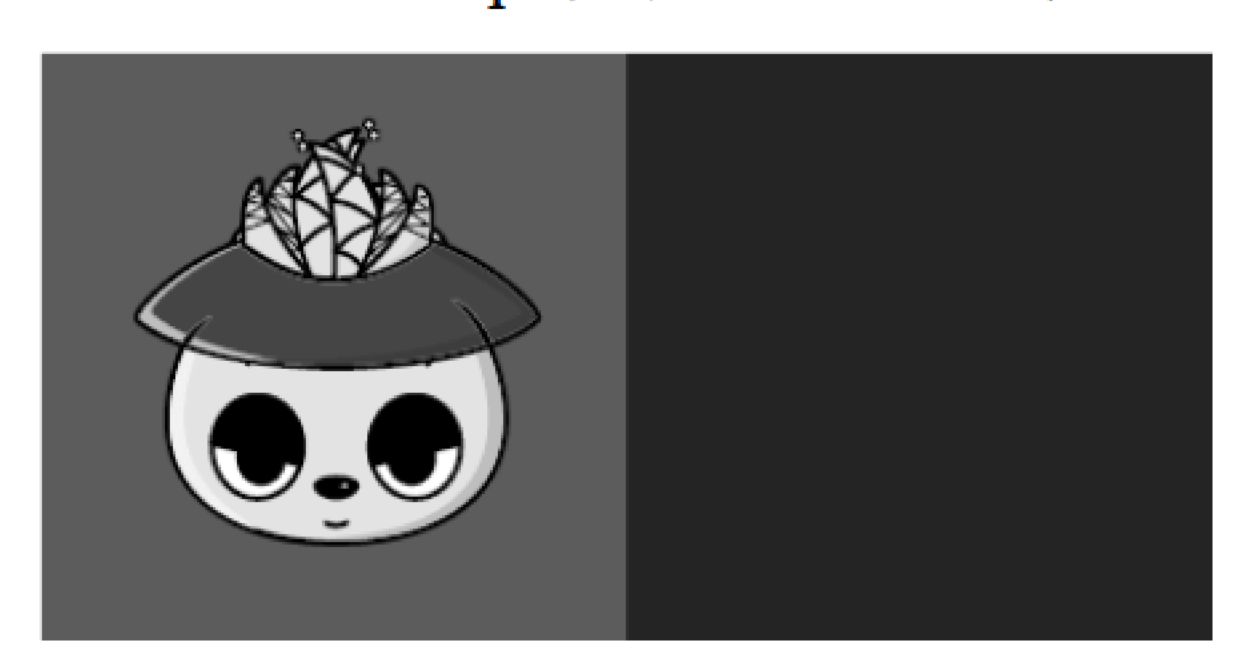


图 13.27　图片位于左侧



图 13.28　图片位于右侧

14. 将图片拖放到回收站　　▷①②③④⑤⑥

编写程序，模拟计算机中将文件拖放到回收站的效果。在页面中有几个图片文件，用鼠标将其拖曳到右侧回收站（垃圾桶）中，即可删除图片，效果如图 13.29 和图 13.30 所示。

（提示：在 ondrop 事件的处理函数中应用 removeChild()方法）

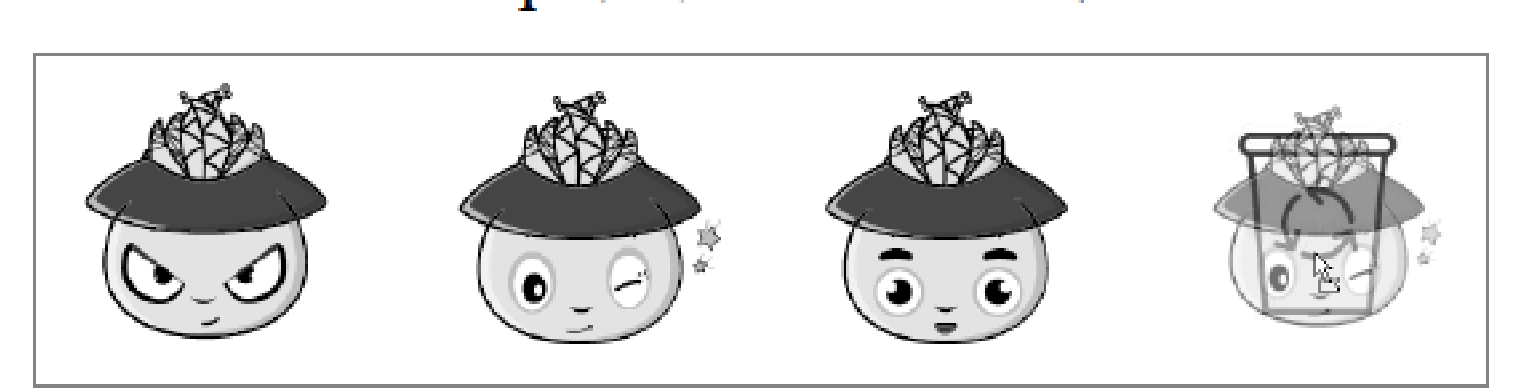


图 13.29　将图片拖动到回收站中

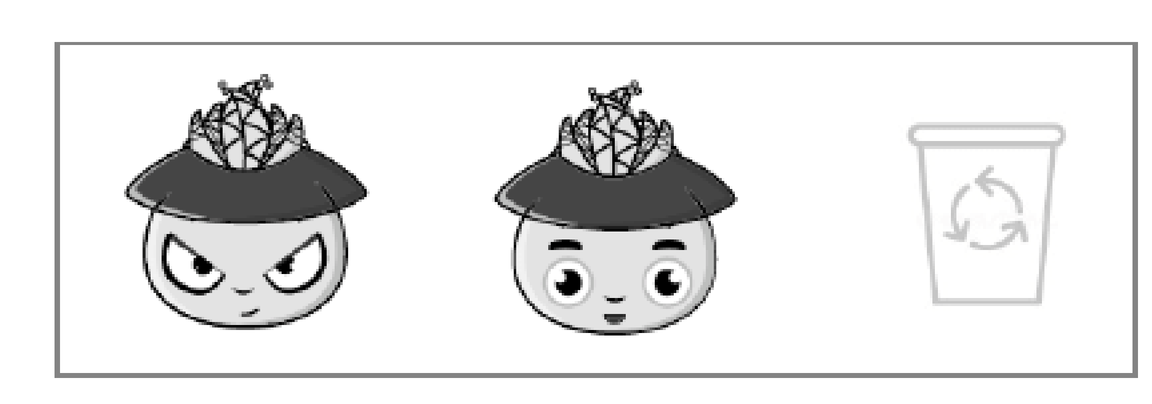
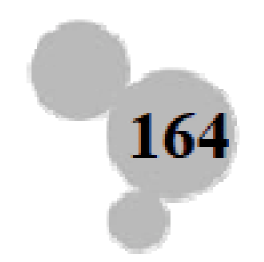


图 13.30　图片已被删除



第 14 章　JavaScript 对象编程



本章训练任务对应核心技术分册第 14 章“JavaScript 对象编程”部分。

重点练习内容：

1. Window窗口对象的应用。
2. 文档前景色和背景色的设置。
3. 使用JavaScript操作表单。
4. DOM节点属性的使用。
5. 对DOM节点的操作。

应用技能拓展学习

1. setInterval()方法和 clearInterval()方法

（1）setInterval()方法

setInterval()方法是 Window 对象中的方法，可按照指定周期（以毫秒计）调用函数或计算表达式。语法格式如下：

```
setInterval(要执行的代码或函数，等待的毫秒数)
```

例如，每隔两秒钟执行一次 JavaScript 代码，在页面中弹出一个对话框。代码如下：

```
<script type="text/javascript">
    setInterval("alert('Hello JS');", 2000);
</script>
```

例如，每隔一秒钟执行一次函数，实时显示当前时间。代码如下：

```
<p id="demo"></p>
<script type="text/javascript">
    var myVar = setInterval(function(){ myTime() }, 1000);
    function myTime() {
        var now = new Date();
        var time = now.toLocaleTimeString();
        document.getElementById("demo").innerHTML = time;
    }
</script>
```

运行结果：上午 10:17:56

（2）clearInterval()方法

clearInterval()方法可取消由 setInterval()方法设定的定时执行操作。该方法的参数必须是由 setInterval()方法返回的 ID 值。语法格式如下：

```
clearInterval(id_of_setinterval)
```

参数 id_of_setinterval 为调用 setInterval()函数时所获得的返回值。

例如，单击“停止”按钮时，取消实时显示当前时间的操作。代码如下：

```
<p id="demo"></p>
<button onclick="stop()">停止</button>
<script type="text/javascript">
    var timerID = setInterval(function(){ myTime() }, 1000);
    function myTime() {
        var now = new Date();
        var time = now.toLocaleTimeString();
        document.getElementById("demo").innerHTML = time;
    }
    function stop() {
        clearInterval(timerID);
    }
</script>
```

2. classList 属性

classList 属性可以获取应用于指定元素的类名集合。该属性用于在元素中添加、移除或切换 CSS 类。语法如下：

```
element.classList
```

参数 element 为要操作的元素。

例如，输出 id 属性值为 demo 的 div 元素的类名，代码如下：

```
<div id="demo" class="a b"></div>
<script type="text/javascript">
    var oDiv = document.getElementById("demo");
    document.write(oDiv.classList);
</script>
```

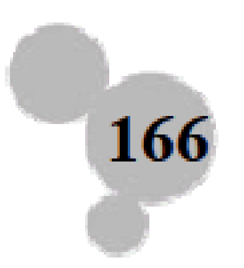
运行结果：a b

classList 属性是只读属性，可以使用 add()和 remove()方法修改该属性。

（1）add()方法

add()方法用于在元素中添加一个或多个类名。如果指定的类名已存在，则不再添加。语法格式如下：

```
element.classList.add(class1, class2, ...)
```



参数 class1、class2 为要添加的类名。

例如，为 id 属性值为 demo 的 div 元素添加两个 CSS 类 c 和 d，并输出添加后元素的类名。代码如下：

```
<div id="demo" class="a b"></div>
<script type="text/javascript">
    var oDiv = document.getElementById("demo");
    oDiv.classList.add("c","d");
    document.write(oDiv.classList);
</script>
```

运行结果：a b c d

（2）remove()方法

remove()方法用于移除元素中的一个或多个类名。语法格式如下：

```
element.classList.remove(class1, class2, ...)
```

参数 class1、class2 为要移除的类名。

例如，为 id 属性值为 demo 的 div 元素移除 3 个 CSS 类 b、c 和 d，并输出移除后元素的类名。代码如下：

```
<div id="demo" class="a b c d"></div>
<script type="text/javascript">
    var oDiv = document.getElementById("demo");
    oDiv.classList.remove("b","c","d");
    document.write(oDiv.classList);
</script>
```

运行结果：a

（3）toggle()方法

toggle()方法用于在元素中切换类名。语法格式如下：

```
element.classList.toggle(class, true|false)
```

☑　class：必选参数，表示要在元素中移除的类名。如果该类名不存在，则在元素中添加类名。

☑　true|false：可选参数，是一个布尔值，用于设置元素是否强制添加或移除类（不管该类名是否存在）。

例如，在 id 属性值为 show 的 div 元素中切换类名，单击该 div 元素时通过切换类名实现元素样式的切换。代码如下：

```
<style type="text/css">
    .oldstyle{
        width: 200px;
        height: 50px;
        line-height: 50px;
        background-color: lightgreen;
        text-align: center;
        color: red;
```

```
        cursor: pointer;
    }
    .newstyle {
        width: 200px;
        height: 50px;
        line-height: 50px;
        background-color: blue;
        text-align: center;
        color: white;
        cursor: pointer;
    }
</style>
<div id="show" class="oldstyle">是金子总会发光</div>
<script type="text/javascript">
    document.getElementById("show").onclick = function (){
        this.classList.toggle("newstyle");
    }
</script>
```

运行结果如图 14.1 和图 14.2 所示。



图 14.1　样式效果（1）

图 14.2　样式效果（2）

3．style 对象的几个属性

（1）backgroundColor 属性

backgroundColor 属性可以设置或返回元素的背景颜色。

设置 backgroundColor 属性的语法格式如下：

```
element.style.backgroundColor="color|inherit|transparent"
```

☑ color：定义背景颜色。

☑ inherit：表示背景颜色从父元素继承。

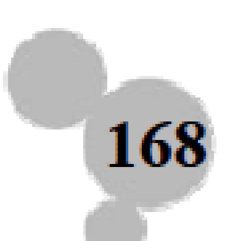
☑ transparent：默认值。表示背景颜色是透明的。

返回 backgroundColor 属性的语法格式如下：

```
element.style.backgroundColor
```

例如，单击“设置背景颜色”按钮时，设置页面背景颜色为绿色，代码如下：

```
<script type="text/javascript">
    function setBg(){
        document.body.style.backgroundColor="green";
    }
</script>
<button type="button" onclick="setBg()">设置背景颜色</button>
```



（2）cursor 属性

cursor 属性可以设置或返回鼠标指针显示的光标类型。

设置 cursor 属性的语法格式如下：

```
Object.style.cursor="value"
```

value 的常用值及其描述如表 14.1 所示。

表 14.1　value 的常用值及其描述

值	光标描述
auto	默认光标，即浏览器设置的光标（通常是一个箭头）
crosshair	十字线
default	默认光标（通常是一个箭头）
move	交叉箭头，指示某对象可被移动
pointer	指示链接的指针（通常是一只手）
wait	程序正忙（通常是一只表或沙漏）

返回 cursor 属性的语法格式如下：

```
Object.style.cursor
```

例如，当鼠标指向文本时设置指针光标类型为手型，代码如下：

```
<p id="demo">路漫漫其修远兮</p>
<script type="text/javascript">
    document.getElementById("demo").style.cursor="pointer";
</script>
```

（3）height 属性

height 属性可以设置或返回元素的高度。

设置 height 属性的语法格式如下：

```
Object.style.height="auto|length|%|inherit"
```

☑　auto：默认。浏览器设置的高度。
☑　length：使用 px、cm 等单位定义高度。
☑　%：定义高度为父元素的百分比。
☑　inherit：height 属性的值从父元素继承。

返回 height 属性的语法格式如下：

```
Object.style.height
```

例如，当单击“改变高度”按钮时设置按钮的高度为 60 像素，代码如下：

```
<script type="text/javascript">
    function changeHeight(){
        document.getElementById("but").style.height="60px";
```

```
    }
</script>
<input type="button" id="but" onclick="changeHeight()" value="改变高度">
```

运行结果如图 14.3 和图 14.4 所示。



图 14.3　实现效果



图 14.4　实现效果

（4）marginTop 属性

marginTop 属性可以设置或返回元素的上外边距。

设置 marginTop 属性的语法格式如下：

```
Object.style.marginTop="%|length|auto|inherit"
```

☑ %：定义基于父元素宽度的百分比上外边距。
☑ length：使用 px、cm 等单位定义上外边距的宽度。
☑ auto：浏览器设定的上外边距。
☑ inherit：上外边距从父元素继承。

返回 marginTop 属性的语法格式如下：

```
Object.style.marginTop
```

例如，设置 id 为 show 的元素的上外边距为 50px，代码如下：

```
<script type="text/javascript">
    document.getElementById("show").style.paddingTop="50px";
</script>
```

（5）backgroundPosition 属性

backgroundPosition 属性可以设置或返回元素背景图像的位置。

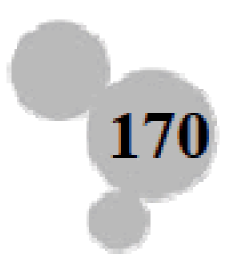
设置 backgroundPosition 属性的语法格式如下：

```
Object.style.backgroundPosition="position"
```

参数 position 的可取值如表 14.2 所示。

表 14.2　参数 position 的可取值

值	描　　述
top left、top center、top right、center left、center center、center right、bottom left、bottom center、bottom right	如果仅指定一个关键字，则另一个值默认为 center
x% y%	x 值表示水平位置，y 值表示垂直位置。左上角是 0% 0%。右下角是 100% 100%。如果仅指定一个值，则另一个值默认为 50%



续表

值	描　述
xpos ypos	x 值表示水平位置，y 值表示垂直位置。左上角是 0 0。单位可以是像素（0px 0px）或任何其他的 CSS 单位。如果仅指定一个值，则另一个值默认为 50%。还可以混合使用%和单位
inherit	背景位置属性的设置从父元素继承

返回 backgroundPosition 属性的语法格式如下：

```
Object.style.backgroundPosition
```

例如，设置 id 属性值为 show 的 div 元素的背景图像位于元素右下角的位置，代码如下：

```
<div id="show"></div>
<script type="text/javascript">
    var myDiv = document.getElementById("show");
    myDiv.style.backgroundPosition="bottom right";
</script>
```

（6）backgroundImage 属性

backgroundImage 属性可以设置或返回元素的背景图像。

设置 backgroundImage 属性的语法格式如下：

```
Object.style.backgroundImage="url('URL')|none|inherit"
```

☑　url('URL')：定义图像文件的位置。
☑　none：默认值，表示没有背景图像。
☑　inherit：该属性的设置从父元素继承。

返回 backgroundImage 属性的语法格式如下：

```
Object.style.backgroundImage
```

例如，为页面设置一个背景图像，代码如下：

```
<script type="text/javascript">
    document.body.style.backgroundImage="url('face.png')";
</script>
```

4．按钮的 disabled 属性

在表单元素的按钮中，disabled 属性用于返回或设置按钮是否被禁用。该属性值为 true 时，禁用按钮；该属性值为 false 时，启用按钮。

例如，设置“添加”按钮禁用，代码如下：

```
<button type="button" id="add">添加</button>
<script type="text/javascript">
    var myButton = document.getElementById("add");
    myButton.disabled = true;
</script>
```

5. 下拉菜单的 selectedIndex 属性

在表单元素的下拉菜单中，selectedIndex 属性用于返回或设置下拉菜单中，当前选中选项在所有下拉菜单选项中的索引。该索引从 0 开始，第一个选项的索引为 0，第二个选项的索引为 1，以此类推。

例如，当改变下拉菜单的选项时，输出当前选中选项的索引，代码如下：

```
<select id="type">
    <option selected>动作电影</option>
    <option>喜剧电影</option>
    <option>科幻电影</option>
</select>
<script type="text/javascript">
    var myType = document.getElementById("type");
    myType.onchange = function(){
        alert(myType.selectedIndex);
    }
</script>
```

6. clientHeight 属性

clientHeight 属性用于返回元素可见区域的高度，以像素计。包括元素的高度和上下内边距的高度，但不包括上下边框的高度。

例如，为 id 属性值为 show 的 div 元素设置样式，并输出该元素可见区域的高度，代码如下：

```
<div id="show" style="width: 100px;height: 50px;padding: 5px;border: 1px solid #000000;"></div>
<script type="text/javascript">
    alert(document.getElementById("show").clientHeight);
</script>
```

运行结果：60

7. setAttribute()和 getAttribute()方法

（1）setAttribute()方法

setAttribute()方法用于创建或改变某个新属性。如果指定属性已存在，则会设置该属性值。语法格式如下：

```
element.setAttribute(attributename,attributevalue)
```

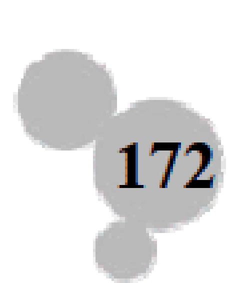
☑ attributename：必选参数，要添加的属性名称。

☑ attributevalue：必选参数，要添加的属性值。

（2）getAttribute()方法

getAttribute()方法用于通过属性名称获取属性的值。语法格式如下：

```
element.getAttribute(attributename)
```



参数 attributename 为必选参数，表示要获取属性值的属性名称。

例如，为 id 属性值为 show 的 div 元素设置 name 属性，并输出该属性的值，代码如下：

```
<div id="show"></div>
<script type="text/javascript">
    var myDiv = document.getElementById("show");
    myDiv.setAttribute("name","music");
    alert(myDiv.getAttribute("name"));
</script>
```

运行结果：music

8. 数组的 splice()方法

splice()方法可以删除数组中指定位置的元素，还可向数组的指定位置添加新元素。语法格式如下：

```
arrayObject.splice(start,length,element1,element2,…)
```

- ☑ arrayObject：必选项，数组名称。
- ☑ start：必选项，指定要删除的数组元素的开始位置，即数组的下标。
- ☑ length：可选项，指定要删除的数组元素的个数。如果未设置该参数，则删除从 start 开始到原数组末尾的所有元素。
- ☑ element：可选项，要添加到数组的新元素。

例如，在 splice()方法中应用不同参数，对相同数组中的元素进行删除操作。代码如下：

```
<script type="text/javascript">
    var arr1 = new Array("a","b","c","d");
    arr1.splice(1);
    document.write(arr1+"<br>");
    var arr2 = new Array("a","b","c","d");
    arr2.splice(1,2);
    document.write(arr2+"<br>");
    var arr3 = new Array("a","b","c","d");
    arr3.splice(1,2,"e","f");
    document.write(arr3+"<br>");
    var arr4 = new Array("a","b","c","d");
    arr4.splice(1,0,"e","f");
    document.write(arr4);
</script>
```

运行结果：a

a,d

a,e,f,d

a,e,f,b,c,d

9. globalCompositeOperation 属性

globalCompositeOperation 是 canvas 中的属性，用于设置或返回如何将一个源（新的）图像绘制到目标（已有的）的图像上。语法格式如下：

```
context.globalCompositeOperation=value;
```

value 的可用值及其描述如表 14.3 所示。

表 14.3　value 的可用值及其描述

值	描　　述
source-over	默认值，在目标图像上显示源图像
source-atop	在目标图像顶部显示源图像。源图像位于目标图像之外的部分是不可见的
source-in	在目标图像中显示源图像。只有目标图像之内的源图像部分会显示，目标图像是透明的
source-out	在目标图像之外显示源图像。只有目标图像之外的源图像部分会显示，目标图像是透明的
destination-over	在源图像上显示目标图像
destination-atop	在源图像顶部显示目标图像。目标图像位于源图像之外的部分是不可见的
destination-in	在源图像中显示目标图像。只有源图像之内的目标图像部分会被显示，源图像是透明的
destination-out	在源图像之外显示目标图像。只有源图像之外的目标图像部分会被显示，源图像是透明的
lighter	显示源图像及目标图像
copy	显示源图像，忽略目标图像
xor	使用异或操作对源图像与目标图像进行组合

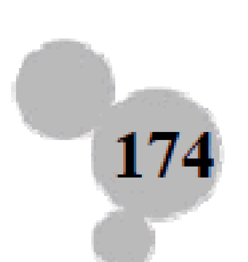
例如，应用 globalCompositeOperation 属性绘制图形。代码如下：

```
<canvas id="myCanvas" width="300" height="150" style="border:1px solid #CCCCCC;"></canvas>
<script type="text/javascript">
    var canvas=document.getElementById("myCanvas");
    var ctx=canvas.getContext("2d");
    ctx.fillStyle="red";
    ctx.fillRect(60,20,150,60);
    ctx.fillStyle="blue";
    ctx.globalCompositeOperation="destination-out";
    ctx.fillRect(90,50,150,60);
</script>
```

实现效果如图 14.5 所示。



图 14.5　绘制图形



实战技能强化训练

训练一：基本功强化训练

1. 删除订单信息　▷①②③④⑤⑥

购物网站的管理员通常都具备删除订单的操作权限。编写程序，在订单信息页面中单击欲删除订单后面的“删除”超链接，将弹出“确定要删除吗？”提示对话框；单击“确定”按钮，将执行删除操作。实现效果如图 14.6 和图 14.7 所示。

（提示：应用 removeChild()方法删除元素）

订单号	商品名称	数量	单价	消费金额	下单日期	操作
13	全自动洗衣机	1	1699	1699	2019-08-10 10:19	删除
14	零基础学JavaScript	2	50	100	2019-08-12 13:16	删除
15	数码相机	1	2599	2599	2019-08-16 15:16	删除

此网页显示：
确定要删除吗？
禁止此页再显示对话框。
确定　取消

图 14.6　删除订单提示

订单号	商品名称	数量	单价	消费金额	下单日期	操作
13	全自动洗衣机	1	1699	1699	2019-08-10 10:19	删除
14	零基础学JavaScript	2	50	100	2019-08-12 13:16	删除

图 14.7　删除订单后

2. 奥运知识问答　▷①②③④⑤⑥

在提示对话框中设计一个选择题，询问用户 2012 年奥运会在哪个城市举办。用户输入答案并单击“确定”按钮后，在页面中会给出答题结果，如图 14.8 和图 14.9 所示。

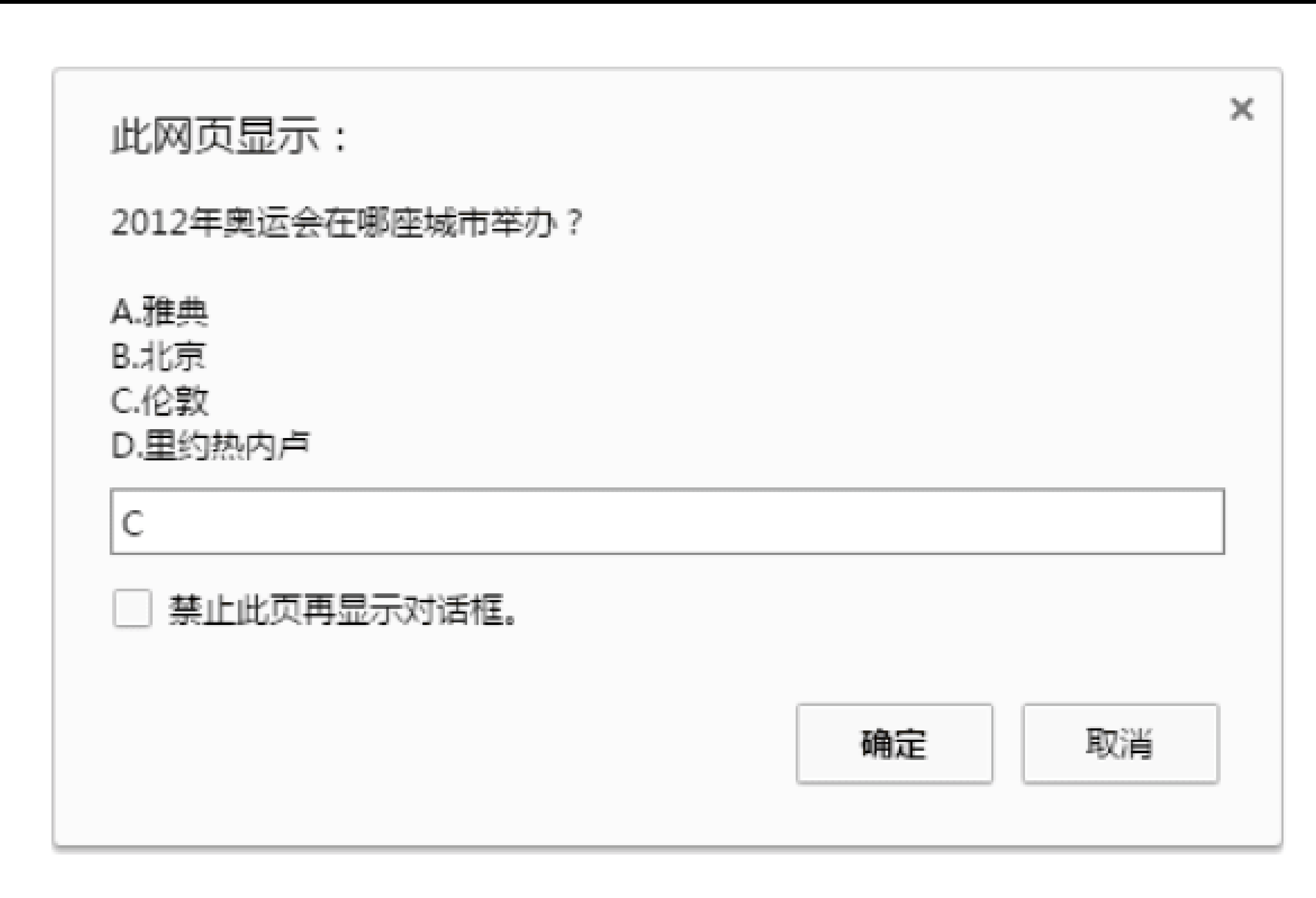


图 14.8　答题

正确！伦敦是2012年奥运会的举办城市

图 14.9　答题结果

3. 单击火箭图片返回顶部 ▷①②③④⑤⑥

实现单击图片快速返回页面顶部功能。向下拖动页面右侧的滚动条，会看到一张火箭图片，单击该图片，页面会快速返回顶部，效果如图 14.10 和图 14.11 所示。

（提示：应用 scrollTo()方法滚动窗口中的文档）

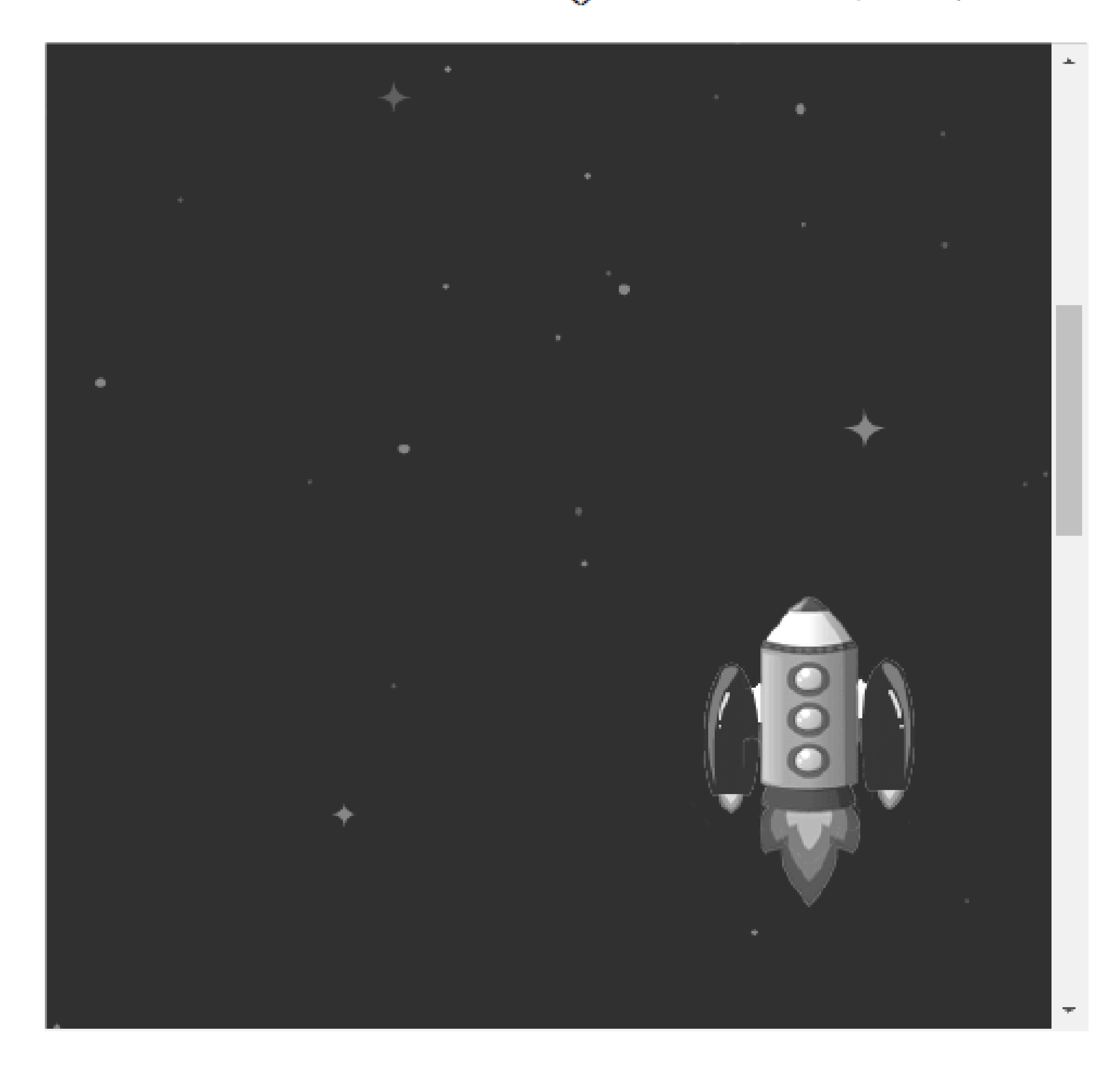


图 14.10　单击火箭前

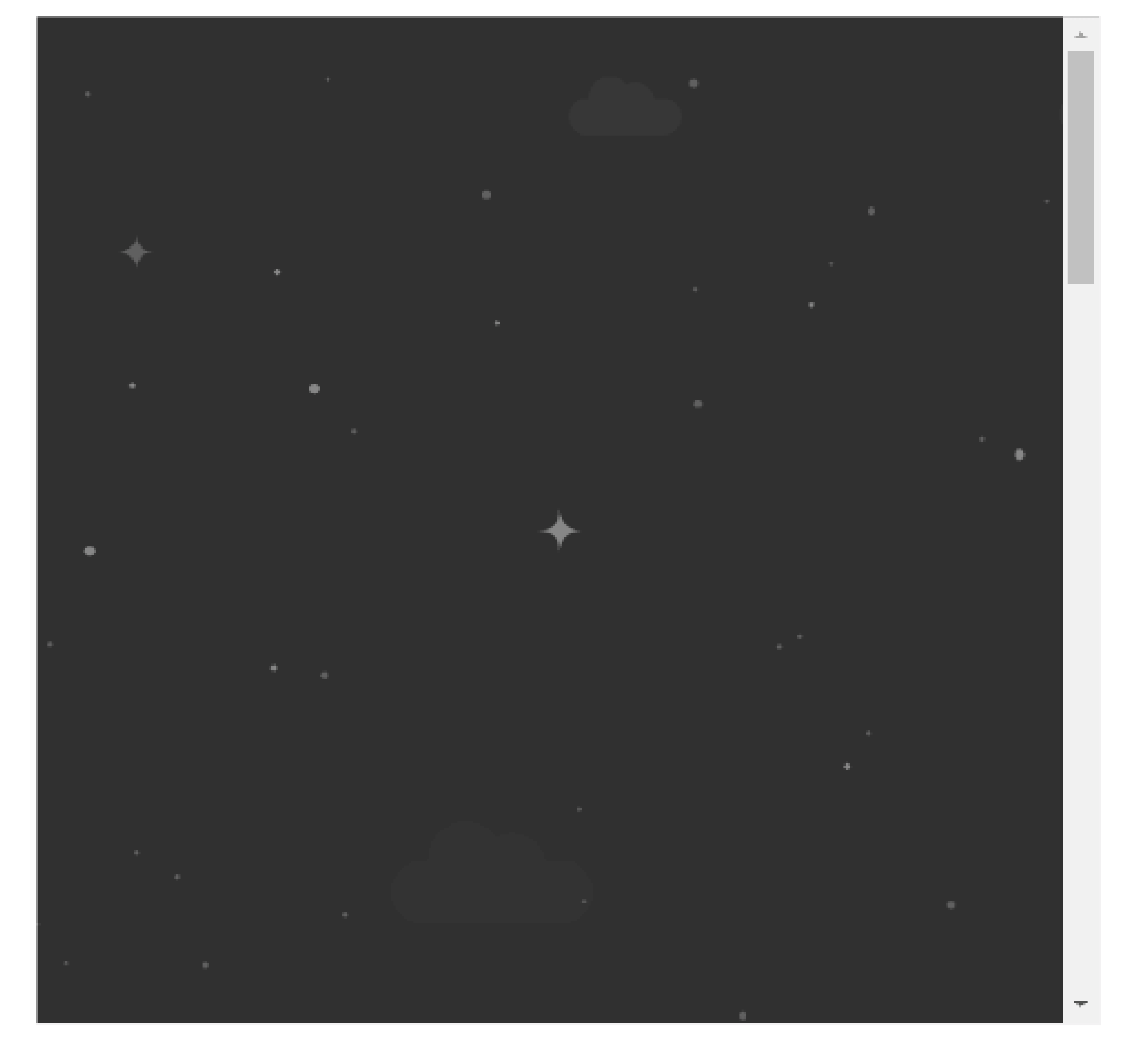


图 14.11　单击火箭后

4. 切换注册按钮的状态 ▷①②③④⑤⑥

注册网站时，只有同意网站的注册协议，才能进一步实现注册。当用户未选中注册协议复选框时，“注册”按钮为禁用状态；当用户选中注册协议复选框时，“注册”按钮为启用状态。编程实现这一功能，效果如图 14.12 和图 14.13 所示。

（提示：应用按钮的 disabled 属性）

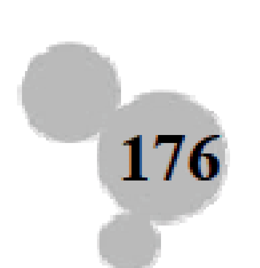




图 14.12　“注册”按钮不可用



图 14.13　“注册”按钮可用

5. 在列表中选择头像　▷①②③④⑤⑥

实现选择用户头像的功能。在“请选择头像”下拉列表框中选择某个头像后，上方将显示该项所代表的头像。用户也可以将该下拉列表框选中，然后按上、下方向键选择想要的头像。实现效果如图 14.14 所示。

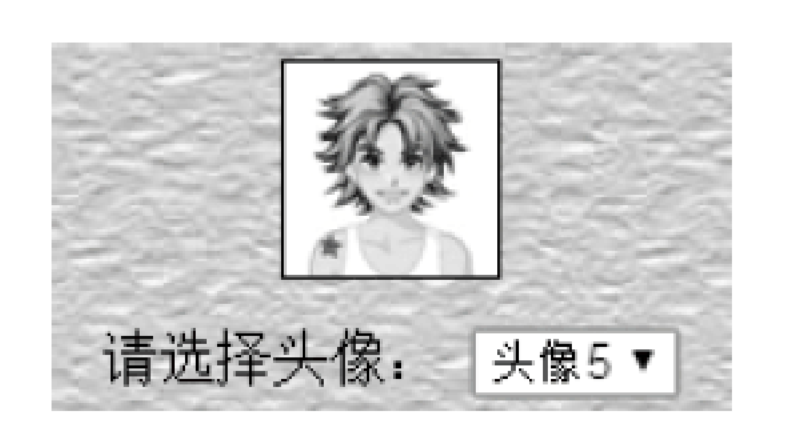


图 14.14　列表中选择头像

6. 使用进度条模拟安装过程　▷①②③④⑤⑥

编写程序，模拟软件安装过程中的进度条效果。当页面加载时，可以看到进度条的完成情况和相应的百分比，当进度条进行到 100%时停止移动，实现效果如图 14.15 和图 14.16 所示。

（提示：在 setInterval()方法中设置进度条元素的宽度）

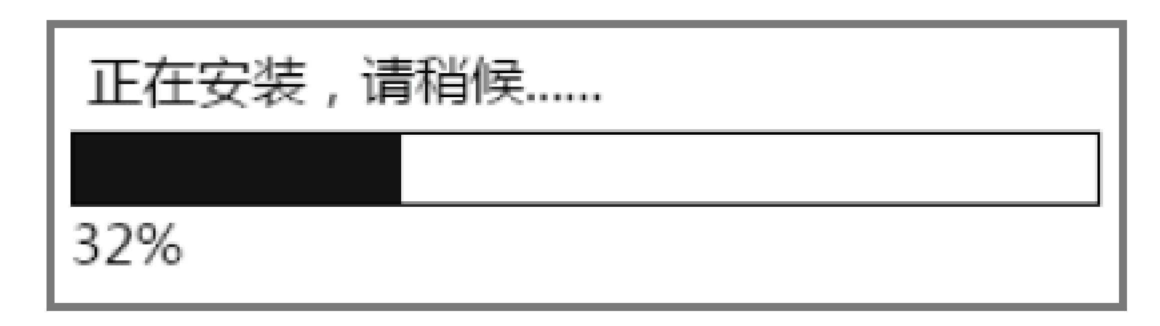


图 14.15　安装中

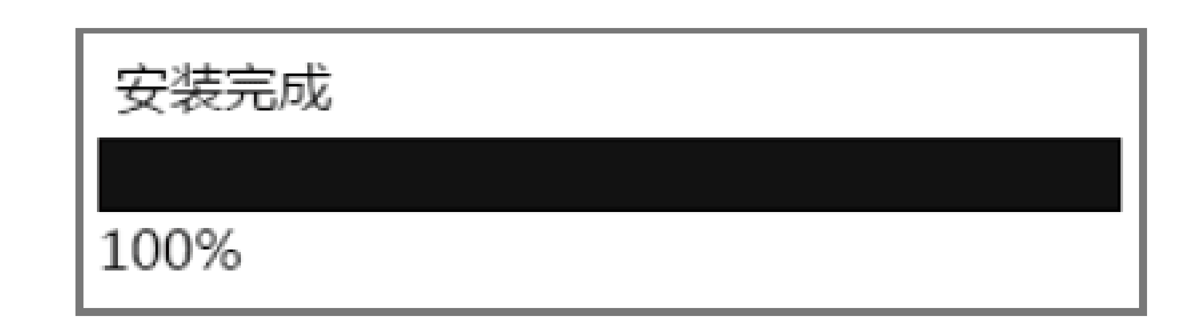


图 14.16　安装完成

7. 实现秒表计时功能　▷①②③④⑤⑥

编写程序，设计一个秒表计时的功能。单击“开始”按钮，开始计时；单击“暂停”按钮暂停计时；再次单击“开始”按钮，继续计时；单击“重置”按钮，使计时器时间归零。实现效果如图 14.17 和图 14.18 所示。

（提示：在“开始”按钮的单击事件中应用 setInterval()方法开始计时，在“暂停”和“重置”按钮的单击事件中应用 clearInterval()方法停止计时）



图 14.17　开始计时

03:63
开始　暂停　重置

图 14.18　暂停计时

8. 输入取票码取票　▷①②③④⑤⑥

星哥在网上买了两张万达影城上映的《美女与野兽》的电影票，电影票的兑换码为 99648500463711。编写程序，模拟自动取票机的取票系统功能，判断星哥取票是否成功，效果如图 14.19 和图 14.20 所示。



图 14.19 取票机取票界面



图 14.20　输入取票码

9. 获取验证码倒计时　▷①②③④⑤⑥

在有些应用中，用户注册需要填写个人的电话号码，并通过短信验证码来完成注册。编写程序，模拟用户注册时获取短信验证码倒计时的功能。当单击“获取验证码”按钮时，按钮文本会以倒计时的形式提示用户重新获取验证码的时间，实现效果如图 14.21 和图 14.22 所示。

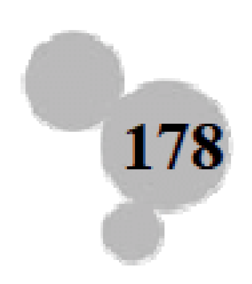
（提示：在 setInterval()方法中设置元素的 innerHTML 属性）



图 14.21　验证码表单



图 14.22　获取验证码中



10．更换页面主题　▷①②③④⑤⑥

设置一个可选择页面主题的下拉列表，当选择某个选项时可以更换主题，实现文档背景色和文本颜色变换的功能，实现效果如图 14.23 和图 14.24 所示。

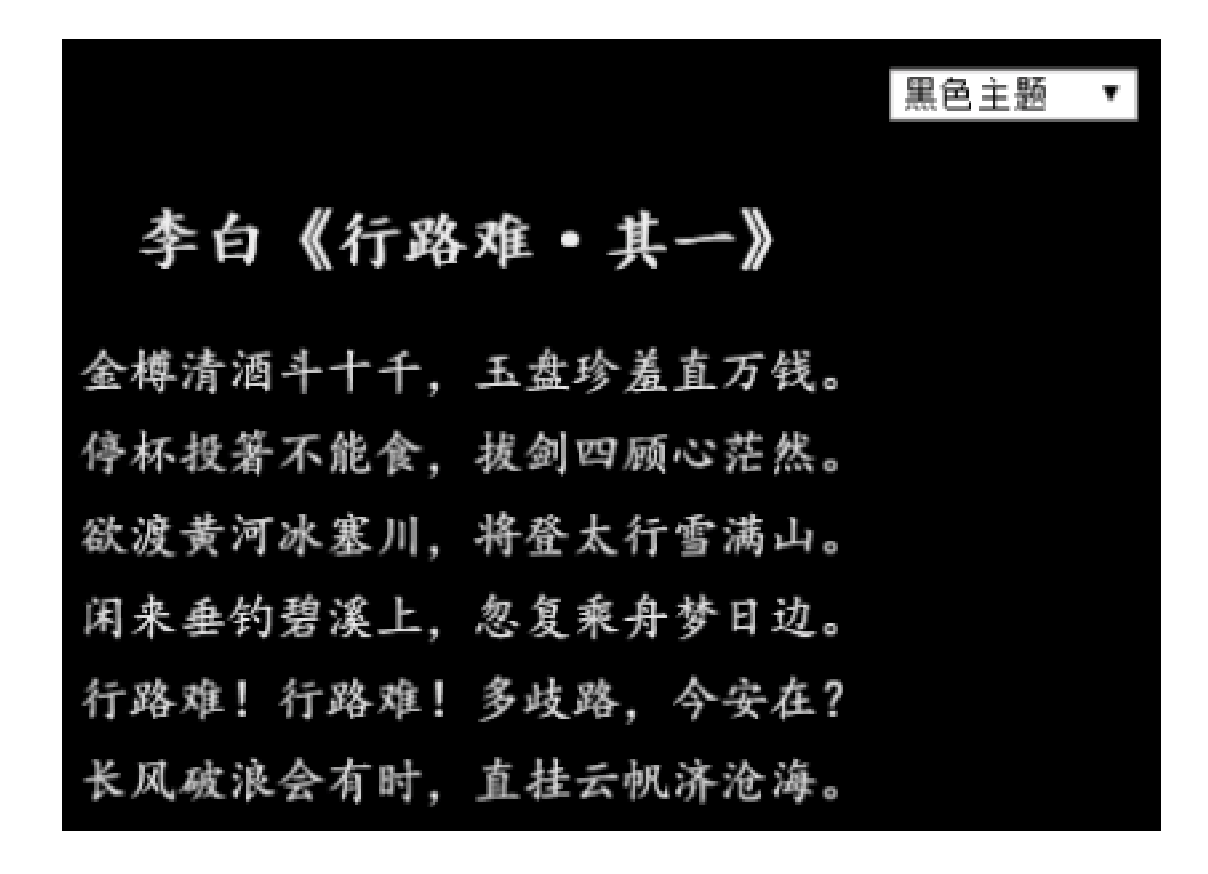


图 14.23　页面主题（1）

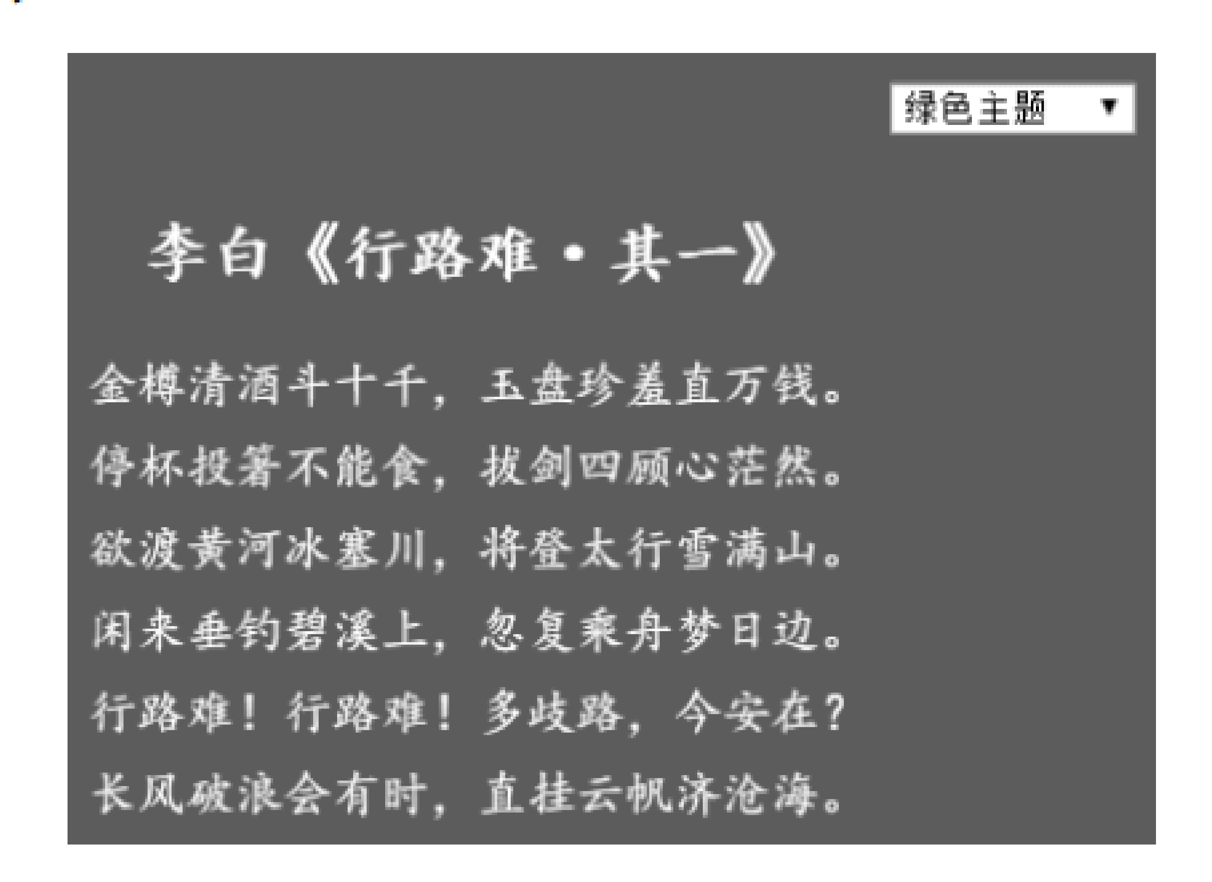


图 14.24　页面主题（2）

训练二：实战能力强化训练

11．实现图片轮播功能　▷①②③④⑤⑥

实现一组具有图片轮播功能的选项卡。页面右侧有 4 个选项卡，每隔 2 秒钟选项卡就会自动切换一次，同时左侧的图片也会相应地随之变化。当鼠标指向某个选项卡时，左侧也会显示对应的图片。实现效果如图 14.25 和图 14.26 所示。

（提示：在 for 语句中为每个选项卡设置一个自定义的 index 属性，作为其索引值）



图 14.25　图一效果



图 14.26　图三效果

12．柱形图显示投票结果　▷①②③④⑤⑥

在投票类网站中，使用柱形图来分析投票结果是比较常用的一种方式。设计一个应用柱形图显示投票结果的功能。在页面中输出 3 个以柱形图表示的投票选项，每个选项的最上方都有一个图片按钮，每单击一次图片按钮，对应的投票数就会加 1，且柱形图的高度会随着投票数的增加而变化，效果如图 14.27 和图 14.28 所示。

（提示：应用 style 对象中的 height 属性设置柱形图的高度）

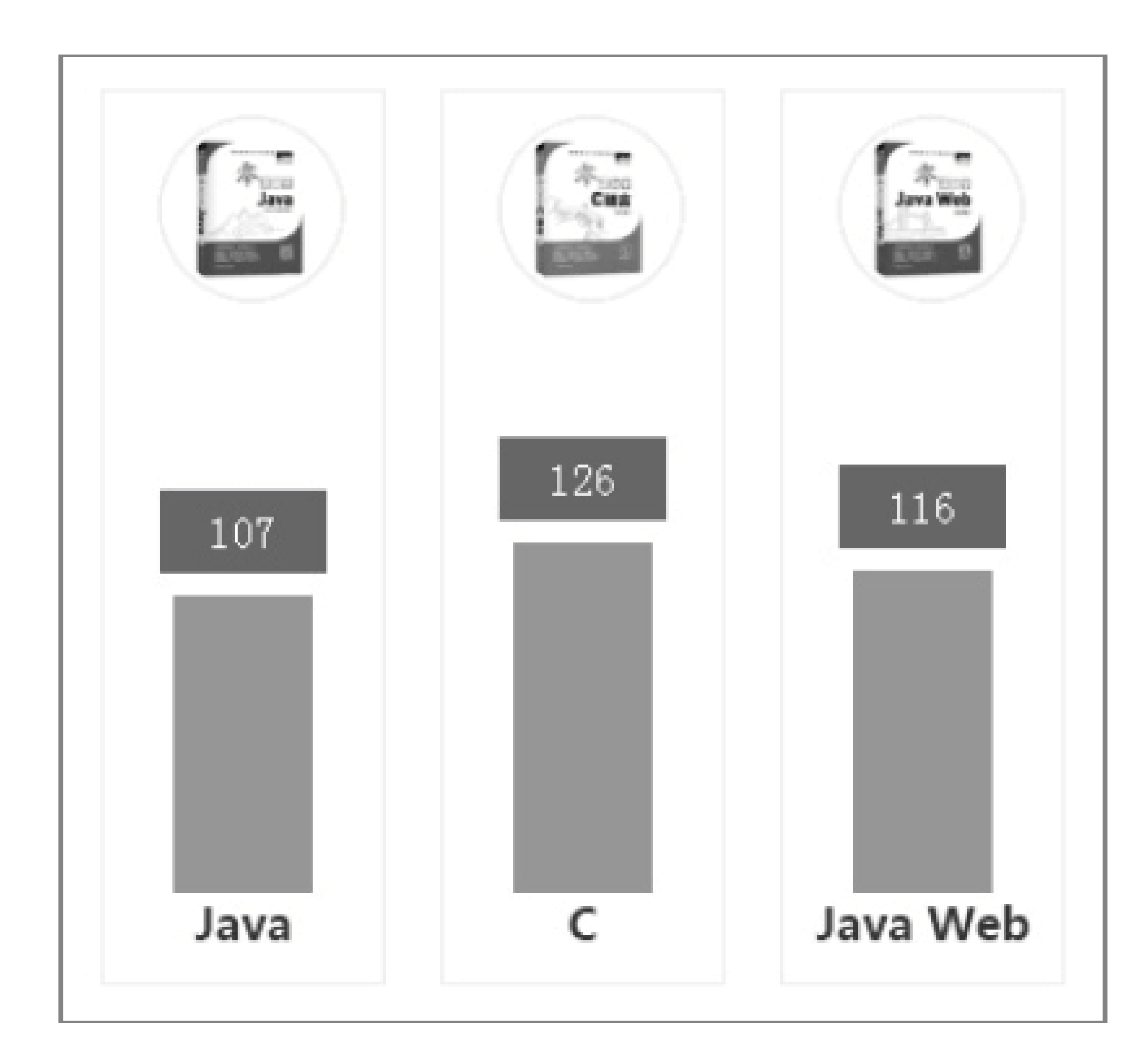


图 14.27　投票统计（1）

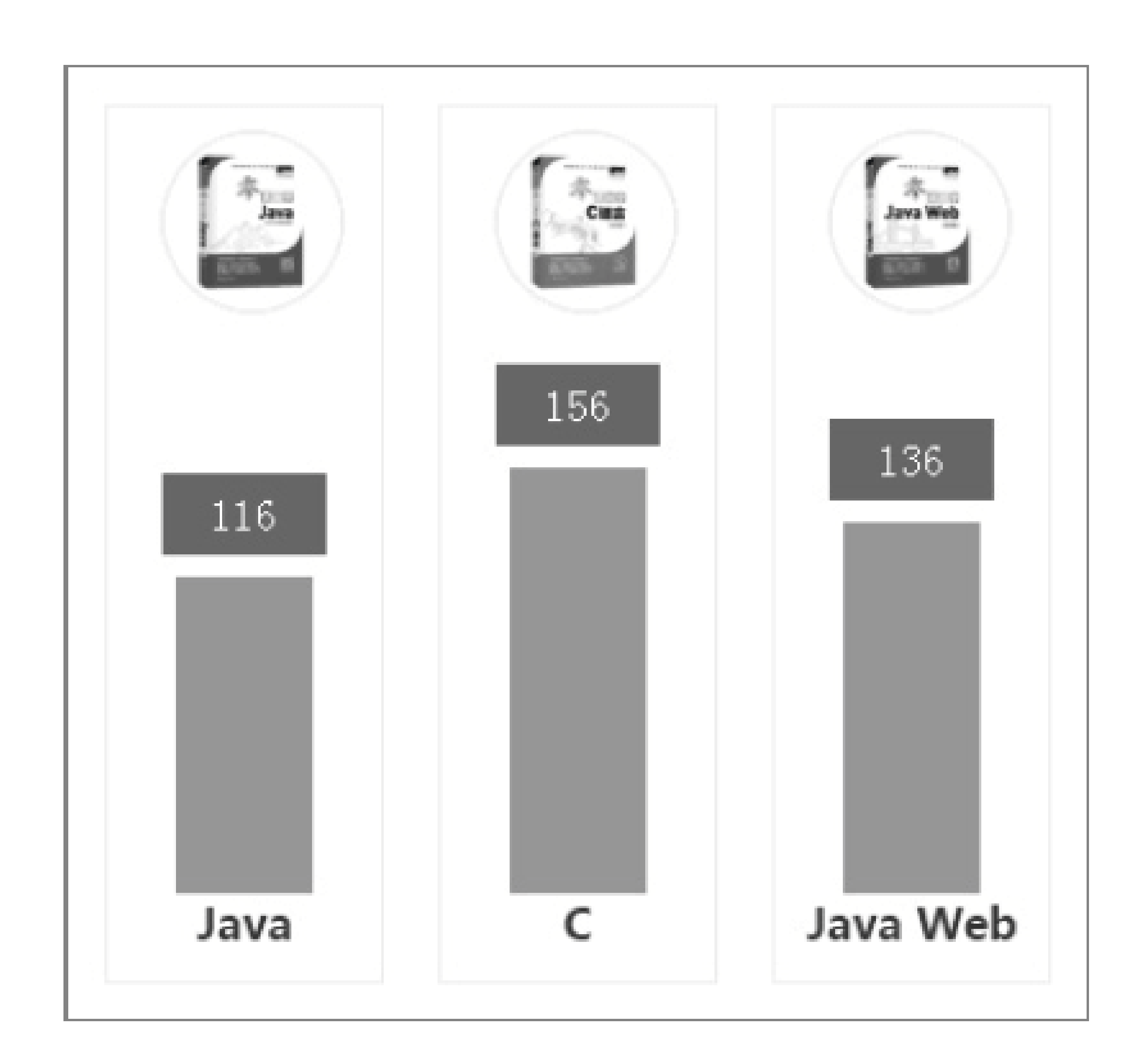


图 14.28　投票统计（2）

13．红心按钮点赞动画特效　▷①②③④⑤⑥

设计一个红心按钮点赞的动画特效。页面中输出 3 个可以点赞的选项，当单击某个心形按钮时，该心形按钮会变成红色，并且在改变颜色的过程中会有一个动画效果。再次单击该心形按钮，可取消点赞，同时心形按钮的颜色会变成灰色。实现效果如图 14.29 和图 14.30 所示。

（提示：应用 classList 属性的 add()方法和 remove()方法为元素添加或移除动画效果）



图 14.29　点赞前



图 14.30　点赞后

14．开心小农场　▷①②③④⑤⑥

操作 DOM 节点，实现一个开心小农场游戏。通过播种、生长、开花和结果 4 个按钮控制作物的生长，实现效果如图 14.31、图 14.32、图 14.33 和图 14.34 所示。

（提示：应用 appendChild()方法添加作物图片，应用 removeChild()方法移除作物图片，应用 replaceChild()方法替换作物图片）



图 14.31　播种



图 14.32　生长

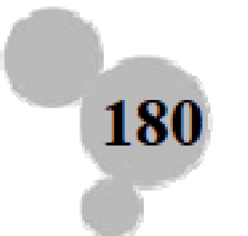





图 14.33　开花



图 14.34　结果

15. 幸运大抽奖

▷①②③④⑤⑥

实现随机抽取幸运用户的抽奖页面。在页面中输出一个用于选择奖项等级的下拉菜单，一个显示参与抽奖用户名称的区域和一个“开始抽奖”按钮。单击“开始抽奖”按钮后，用户名称会随机进行变化，这时下方按钮中的文字变成了“结束抽奖”。单击“结束抽奖”按钮后，用户名称区域会显示中奖用户的名称，同时页面上方的中奖用户头像也会同步进行闪烁，实现效果如图 14.35、图 14.36 和图 14.37 所示。

（提示：应用 random()方法生成的随机数作为用户数组的索引，应用 setInterval()和 clearInterval()方法实现用户名称的变化和停止）



图 14.35　抽奖前



图 14.36　抽奖中



图 14.37　抽奖后

16. 模拟刮刮卡刮奖特效

▷①②③④⑤⑥

为了促进消费和回馈消费者，商家经常会发放有奖刮刮卡。编写程序，模拟刮刮卡的刮奖效果。在页面中有一个矩形的刮卡区域，按住鼠标在刮卡区域内移动，刮掉涂层后会显示中奖结果，实现效果如图 14.38 和图 14.39 所示。

（提示：在 onmousemove 事件的处理函数中，应用 canvas 绘制指定半径的圆形）



图 14.38　显示刮刮卡区域

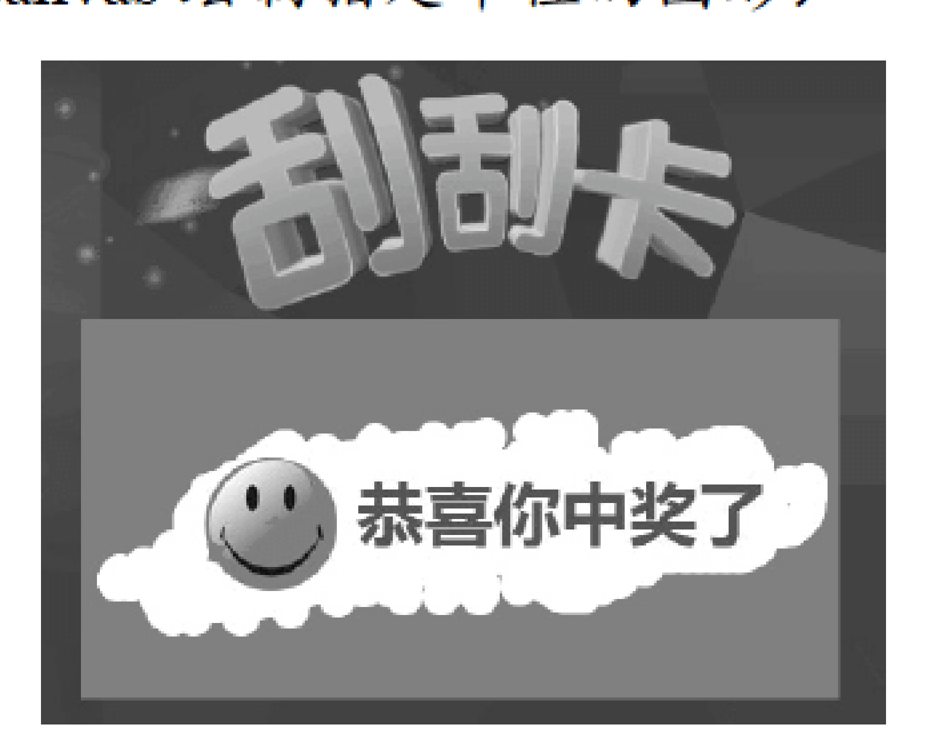


图 14.39　显示中奖结果

17. 歌曲置顶和删除

▷①②③④⑤⑥

模拟点歌系统中置顶歌曲和删除歌曲的功能。单击歌曲和歌手名称右侧的“置顶”按钮可置顶该歌曲；单击歌曲和歌手名称右侧的“删除”按钮，可删除该歌曲。实现效果如图 14.40、图 14.41 和图 14.42 所示。

（提示：应用 insertBefore()方法实现歌曲置顶，应用 removeChild()方法实现歌曲删除）



图 14.40　原始歌单



图 14.41　置顶歌曲

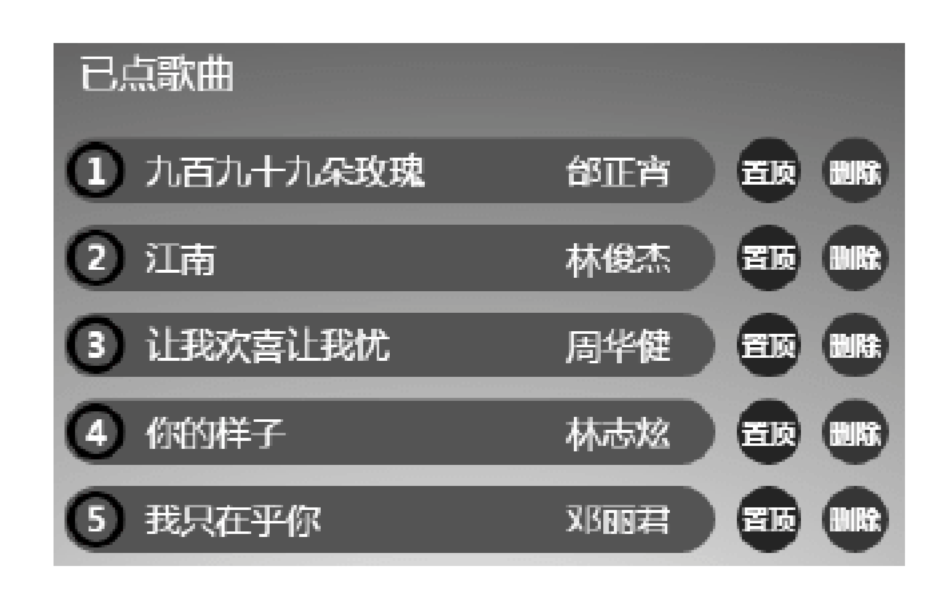


图 14.42　删除歌曲

18. 模拟老虎机滚动抽奖效果

▷①②③④⑤⑥

模拟老虎机的滚动抽奖效果。页面中显示一个抽奖区域，单击下方的“立即抽奖”按钮，抽奖区中的奖品会上下翻滚。停止滚动后，抽奖区显示的奖品即为中奖奖品，效果如图 14.43 和图 14.44 所示。

（提示：设置元素 style 对象中的 top 属性，实现奖品上下滚动）



图 14.43　未中奖



图 14.44　中奖效果

19. 幕帘动画效果

▷①②③④⑤⑥

拉开和关闭幕帘的动画效果可以用于广告特效，也可以用于个人主页。编写程序，实现拉开和关闭幕帘的动画效果。在初始状态下，幕帘是关闭的。单击“拉开幕帘”超链接，幕帘会向左右拉开；单击“关闭幕帘”超链接，幕帘会向中间关闭，实现效果如图 14.45 和图 14.46 所示。

（提示：在 setInterval()方法的处理函数中设置左、右两侧幕帘的宽度）

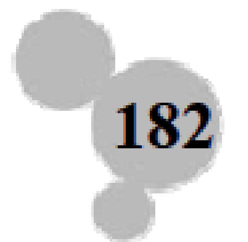




图 14.45　帘幕关闭



图 14.46　帘幕已拉开

20．模拟微信弹出菜单　▷①②③④⑤⑥

模拟微信弹出菜单功能。页面底部显示 3 个主菜单项，单击某菜单项，弹出其对应的子菜单。再次单击其他主菜单项，已弹出的子菜单项被隐藏，同时弹出新的子菜单项，效果如图 14.47 和图 14.48 所示。

（提示：设置元素 style 对象的 top 属性，隐藏和显示元素）



图 14.47　微信菜单



图 14.48　单击弹出子菜单

21．模拟微信输入支付密码　▷①②③④⑤⑥

模拟微信支付时输入支付密码的功能。手机充值页面中有一个“立即支付”按钮，单击该按钮时，页面中弹出包括密码输入框和数字键盘的密码输入界面。按下某个数字按键，选择的数字会以密码形式显示在上方输入框中，当输入的密码为123456时，页面弹出支付成功的提示信息。实现效果如图14.49和图14.50所示。

（提示：设置元素 style 对象的 height 属性，改变高度）

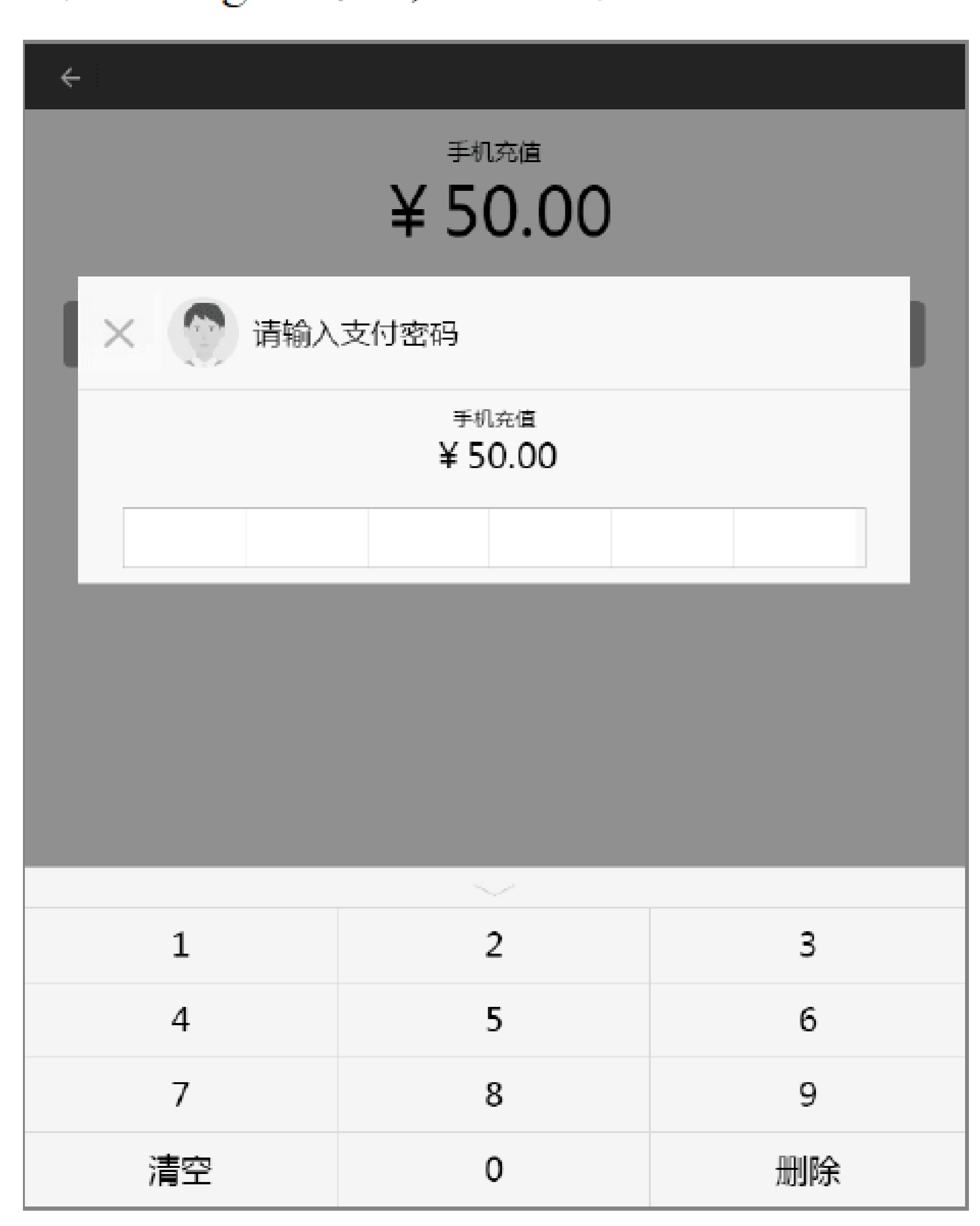


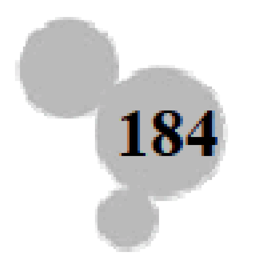
图 14.49　密码输入页面



图 14.50　密码正确后立即支付

22．实现电影海报轮播　▷①②③④⑤⑥

实现热映电影海报轮播的动态效果。每屏显示5张电影海报，单击右上角的向左按钮和向右按钮，会切换至上一屏或下一屏的5部电影海报，效果如图14.51所示。



（提示：在 setInterval()方法的处理函数中应用元素 style 对象的 left 属性）

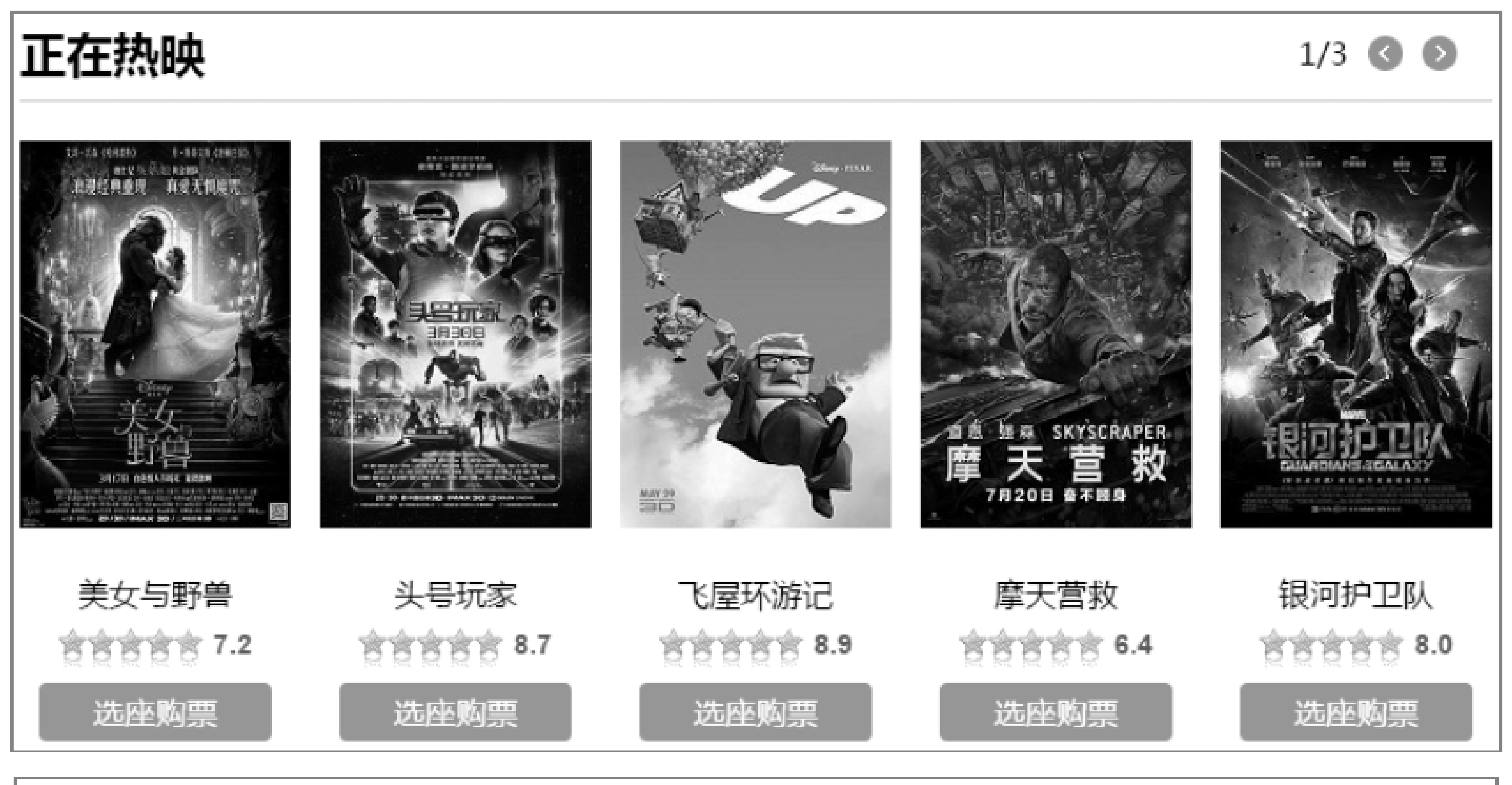




图 14.51　实现效果



第 15 章　响应式网页设计

本章训练任务对应核心技术分册第 15～16 章“响应式网页设计”与“组件”部分。

重点练习内容：

1. 常见响应式布局的实现方式。
2. 各种响应式布局的运用及优缺点。
3. 媒体查询的使用。
4. 运用响应式插件进行响应式布局。

应用技能拓展学习

1. 媒体查询的使用

媒体查询可以根据设备显示器的特性（如视口宽度、屏幕比例和设备方向），设定 CSS 的样式。媒体查询由媒体类型和一个或多个检测媒体特性的条件表达式组成。媒体查询中可用于检测的媒体特性有 width、height 和 color 等。使用媒体查询，可以在不改变页面内容的情况下，为特定的输出设备定制显示效果。

媒体查询的使用步骤如下：

首先，在 HTML 页面的<head>标签中添加 viewport 属性代码。代码如下：

```
<meta name="viewport content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=no"/>
```

其中，各属性值表示的含义如表 15.1 所示。

表 15.1　viewport 属性中常用的属性值及含义

属　性　值	含　　义
width=device-width	设定布局视口宽度
initial-scale=1	设定布局视口高度
maximum-scale=1	设定页面初始缩放比例（0～10）
user-scalable=no	设定用户是否可以缩放（yes/no）

然后，使用@media 关键字编写 CSS 媒体查询代码。

例如，当设备屏幕宽度为 320～720 像素时，在媒体查询中设置 body 的背景色 background-color 的属性值为 red，并覆盖原来的 body 背景色；当设备屏幕宽度小于等于 320 像素时，在媒体查询中设置 body 背景色的 background-color 属性值为 blue，并覆盖原来的 body 背景色。代码如下：

```
/*当设备宽度在 320～720px 时*/
@media screen and (max-width: 720px) and (min-width: 320px) {
    body {
        background-color: red;
    }
}
/*当设备宽度小于等于 320px 时*/
@media screen and (max-width: 320px) {
    body {
        background-color: blue;
    }
}
```

2. 响应式插件之栅格系统

栅格系统（Grid Systems）又叫作网格系统，即运用固定的格子来设计版面布局，其风格工整简洁。具体地说，就是将网页按宽度分为 12 等份，开发人员可以自由分配项目中列所占的份数。例如，定义每列的宽度为 2 格，则一行显示 6 列项目；定义每列的宽度为 3 格，则一行显示 4 列项目；以此类推，网格效果如图 15.1 所示。

这并不表示所有列的总宽度必须填满 12 列，而是只要不超过 12 列就可以。如果超过了 12 列，则自动进行换行处理。

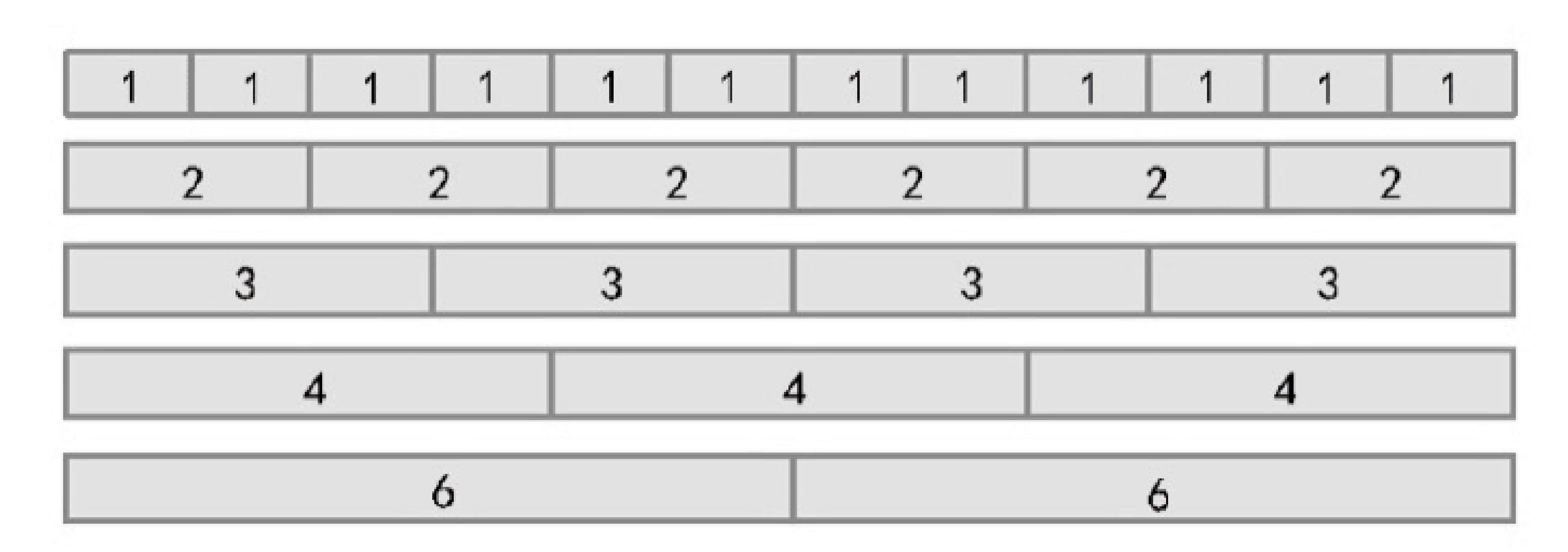


图 15.1　栅格系统

（1）网格化选项

栅格系统提供了 5 个网格等级，每个响应式分界点分隔出一个等级。各等级的屏幕尺寸以及类名前缀如表 15.2 所示。

表 15.2　栅格系统的网格等级

	超小屏幕（<576px）	小屏幕（≥576px）	中等屏（≥768px）	大屏幕（≥992px）	超大屏幕（≥1200px）
.container 最大宽度	None(auto)	540px	720px	960px	1140px
类前缀	.col-	.col-sm-	.col-md-	.col-lg-	.col-md-
适用设备	手机	平板	桌面显示器	大型桌面显示器	超大型桌面显示器

网格布局断点的媒体查询，都是基于宽度最小值（min-width）的。这意味着，对应的媒体查询可应用到该等级之上的所有设备中。例如，.col-md-4 的定义，可以在中等屏幕、大屏幕以及超大屏幕上呈现效果，但在小屏幕和超小屏幕上不会起作用。

（2）设置列的宽度

Bootstrap 3 中，需要严格定义栅格系统中每一列的宽度，如 col-md-3 等。Bootstrap 4 中，由于可与弹性布局相结合，因此可通过以下几种方式定义列的宽度。

☑ col：若一个 row 中定义了多个 col，则这几个 col 平分 row 的宽度。

☑ col-auto：该类名表示当空间缩小时，为该项保留足够空间。

☑ col-{sm|md|xm|lg } -x：该类名表示在对应屏幕范围类占据的个数，

注意：col 类名必须作为 row 的直接子元素，否则宽度设置无效。

实战技能强化训练

训练一：基本功强化训练

1．开心消消乐“最新活动”页面 ▷①②③④⑤⑥

开心消消乐是一款老少皆宜的休闲游戏，并且不时会推出一些活动。通过弹性布局实现开心消消乐官网的“最新活动”模块，如图 15.2 所示。该模块始终在页面中垂直居中显示，放大浏览器时，最左侧的内容放大，而中间和右边内容不变。

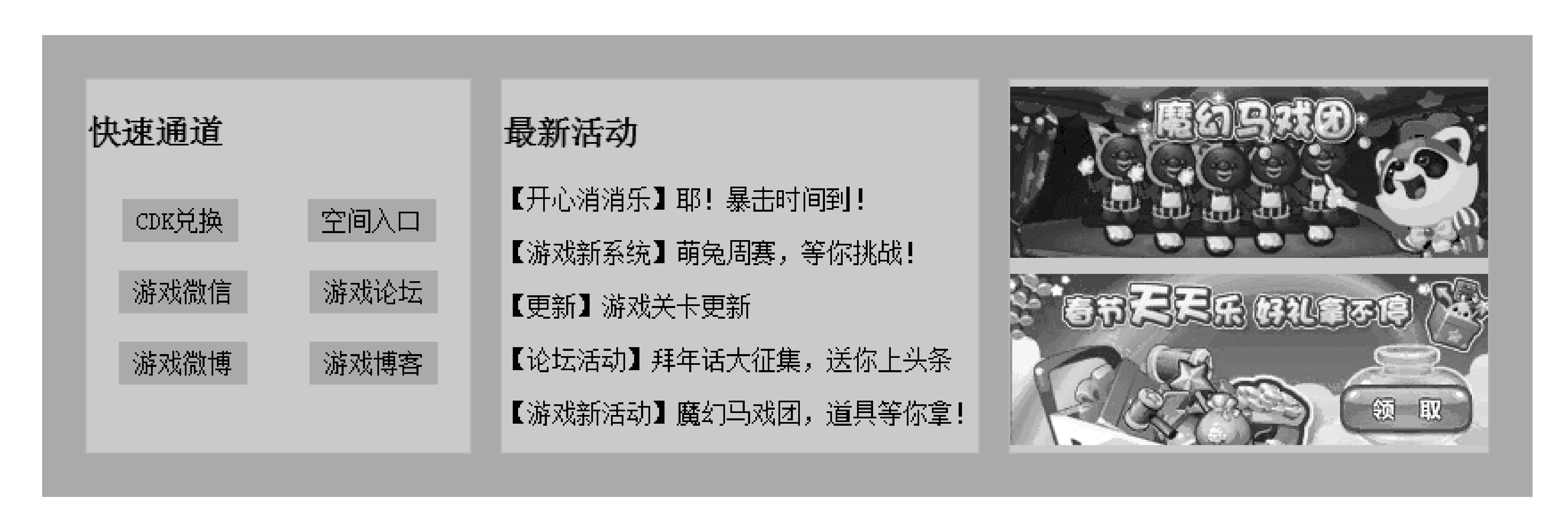
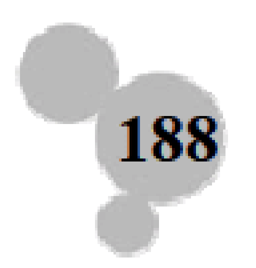


图 15.2 “最新活动”模块

2．浏览器缩小时，隐藏成绩表中的列 ▷①②③④⑤⑥

响应式表格中的一种表现方式是：可根据浏览器屏幕大小隐藏表格中的部分列。实现如图 15.3 所示的考试成绩表，当浏览器缩小至 800 像素时，表格的最后一列被隐藏；当浏览器继续缩小至 640 像素时，表格再次隐藏最后一列，效果如图 15.4 所示。



考试成绩

年份	数学	语文	英语	化学	物理	生物
2017年	85	90	75	68	66	87
2016年	78	96	78	68	78	85
2015年	96	86	93	71	82	93
2014年	74	85	68	63	66	96
2013年	85	90	75	68	66	87
2012年	89	74	85	92	82	74

图 15.3　完全显示的成绩表

考试成绩

年份	数学	语文	英语	化学
2017年	85	90	75	68
2016年	78	96	78	68
2015年	96	86	93	71
2014年	74	85	68	63
2013年	85	90	75	68
2012年	89	74	85	92

图 15.4　表格后两列被隐藏

3．翻转并滚动显示招聘表　▷①②③④⑤⑥

实现响应式表格的方法有 3 种：隐藏、滚动或转换表格中的列。编写程序，当表格完全显示时，滚动显示表格中的列，效果如图 15.5 所示；当浏览器屏幕缩小时，表格中的行、列互换，且表文部分滚动显示，效果如图 15.6 所示。

（提示：要使表格的行、列翻转，需将表格元素的 display 属性强行修改为 block）

工作地点	web前端	Android开发	Python	Java	测试
上海	5人	3人	1人	8人	2人
北京	7人	3人	8人	5人	2人
重庆	5人	5人	7人	2人	20人
广州	4人	6人	9人	5人	3人
深圳	5人	5人	4人	2人	1人
大连	7人	12人	3人	5人	6人

图 15.5　完全显示的招聘表

工作地点	上海	北京	重庆	广州	深圳
web前端	5人	7人	5人	4人	5人
Android开发	3人	3人	5人	6人	5人
Python	1人	8人	7人	9人	4人
Java	8人	5人	2人	5人	2人
测试	2人	2人	20人	3人	1人

图 15.6　行列翻转且滚动显示的表格

4．CSS 实现响应式导航栏 ▷①②③④⑤⑥

通过 CSS 中的媒体查询，实现响应式导航栏。在 PC 端浏览页面时，效果如图 15.7 所示；在手机端浏览页面时，轮播图将隐藏，导航栏由横向变为纵向，运行效果如图 15.8 所示。



图 15.7 PC 端页面效果



图 15.8 移动端页面效果

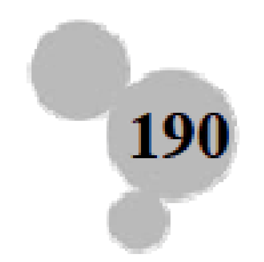
5．实现课程推荐列表页面 ▷①②③④⑤⑥

编写程序，通过 bootstrap 中的弹性布局实现直播精品课程推荐页面。网页缩小或放大时，课程信息会随之伸缩，实现效果如图 15.9 所示。

（提示：通过 flex-grow-1 实现网页伸展，扩展该项目）



图 15.9 直播课程推荐列表



训练二：实战能力强化训练

6. 360 趣玩页面　▷①②③④⑤⑥

通过 bootstrap 实现一则 360 趣玩推送消息。在 PC 端，页面在网页中居中固定显示；在移动端，页面全屏显示。具体实现效果如图 15.10 所示。



图 15.10　趣玩页面

7. 响应式游戏活动介绍页面　▷①②③④⑤⑥

通过 bootstrap 实现响应式游戏活动介绍页面。在 PC 端的运行效果如图 15.11 所示，在移动端的运行效果如图 15.12 所示。

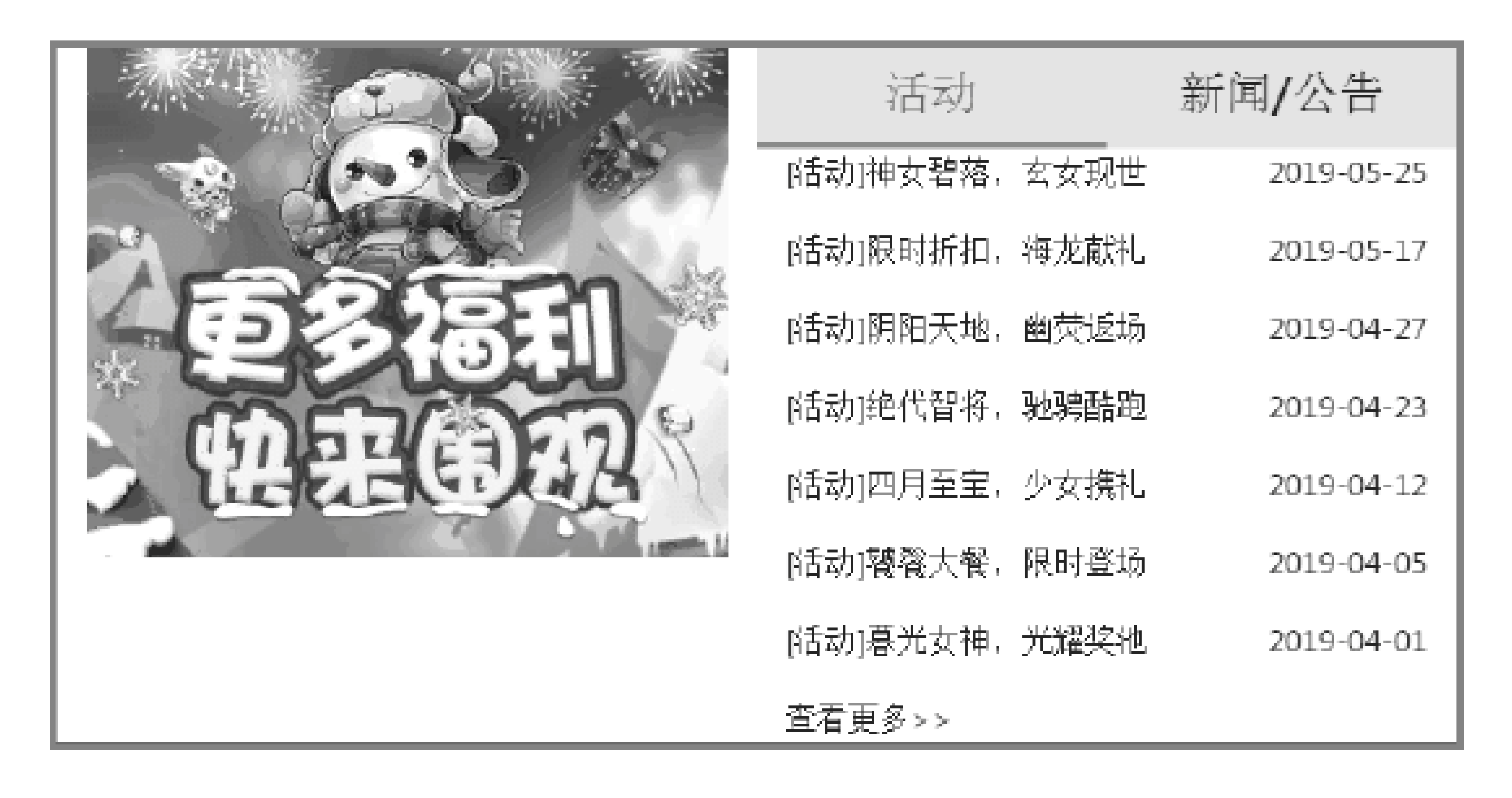


图 15.11　PC 端游戏活动页面



图 15.12　移动端游戏活动页面

8．响应式游戏列表页面　▷①②③④⑤⑥

使用 bootstrap 中的混合布局，实现一个响应式游戏列表页面。初始 PC 端游戏列表显示为 6 列；缩小浏览器屏幕时，游戏列表的列数逐渐减少，如图 15.13 所示；当浏览器屏幕缩小至手机宽度时，游戏列表分两列展示，如图 15.4 所示。



图 15.13　页面分 3 列显示



图 15.14　页面分两列显示

9．女装专场活动页面　▷①②③④⑤⑥

每逢节假日，京东、淘宝等各大购物网站都会开启打折促销活动。为电商网站制作一个女装专场页面，效果如图 15.15 所示。

（提示：通过 col-auto 属性实现元素始终在页面中留有足够空间）

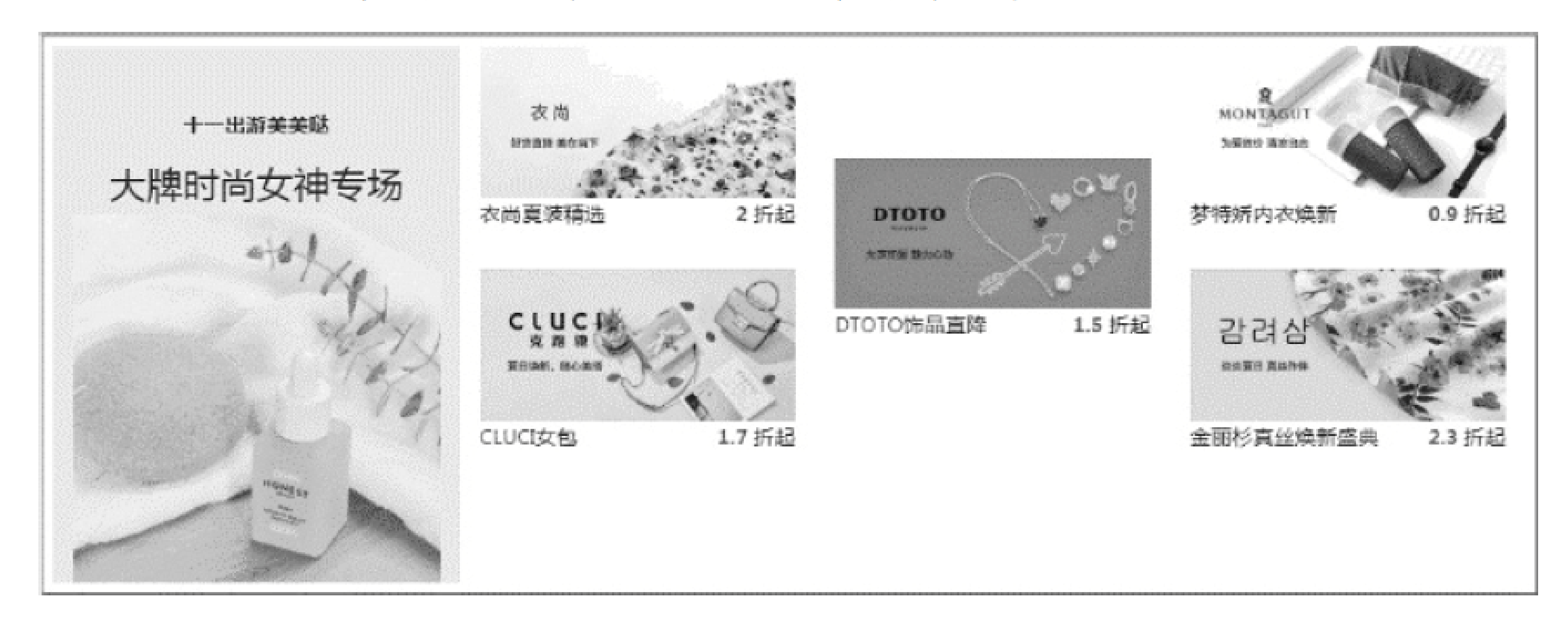
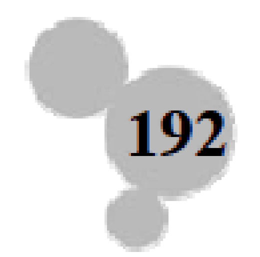


图 15.15　女装专场活动页面



10. Bootstrap 实现图片轮播　▷①②③④⑤⑥

轮播广告是网站重要的一部分。运用 Bootstrap 中的轮播图插件，实现页面图片的轮播效果，如图 15.16 所示。



图 15.16　bootstrap 实现图片轮播

答 案 提 示

第 1 章 基本功训练	第 1 章 实战强化训练	第 2 章 基本功训练	第 2 章 实战强化训练	第 3 章 基本功训练	第 3 章 实战强化训练
第 4 章 基本功训练	第 4 章 实战强化训练	第 5 章 基本功训练	第 5 章 实战强化训练	第 6 章 基本功训练	第 6 章 实战强化训练
第 7 章 基本功训练	第 7 章 实战强化训练	第 8 章 基本功训练	第 8 章 实战强化训练	第 9 章 基本功训练	第 9 章 实战强化训练
第 10 章 基本功训练	第 10 章 实战强化训练	第 11 章 基本功训练	第 11 章 实战强化训练	第 12 章 基本功训练	第 12 章 实战强化训练
第 13 章 基本功训练	第 13 章 实战强化训练	第 14 章 基本功训练	第 14 章 实战强化训练	第 15 章 基本功训练	第 15 章 实战强化训练